

Traffic Predictive Control from Low-Rank Structure

Samuel Coogan^a, Christopher Flores^b, Pravin Varaiya^c

^a*Electrical Engineering Department, University of California, Los Angeles*

^b*Sensys Networks, Inc.*

^c*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley*

Abstract

The operation of most signalized intersections is governed by predefined timing plans that are applied during specified times of the day. These plans are designed to accommodate average conditions and are unable to respond to large deviations in traffic flow. We propose a control approach that adjusts time-of-day signaling plans based on a prediction of future traffic flow. The prediction algorithm identifies correlated, low rank structure in historical measurement data and predicts future traffic flow from real-time measurements by determining which structural trends are prominent in the measurements. From this prediction, the controller then determines the optimal time of day to apply new timing plans. We demonstrate the potential benefits of this approach using eight months of high resolution data collected at an intersection in Beaufort, South Carolina.

1. Introduction

The advent of ubiquitous traffic sensing provides unprecedented real-time, high-resolution data that elucidate historical trends and current traffic conditions. However, traditional signal timing approaches have yet to take full advantage of these data [1]. Surveys of practitioners suggest that only sixty percent of the 300,000 signalized intersections in the United States are retimed at intervals less than five years [2], and the National Transportation Operations Coalition has given a grade of “C–” to signal timing practices and a grade of “F” to traffic monitoring and data collection in the United States. These deficiencies contributed to the 6.9 billion hours of additional travel time caused by inefficient traffic management in 2015 [3].

Increased urbanization demands more efficient use of this transportation infrastructure, which in turn requires full use of measured data. At signalized intersections, standard actuated signal timing plans are designed to make limited use of real-time measurements to, *e.g.*, extend green time for approaching vehicles or enable actuation phases to be skipped if no waiting vehicles are present [4]. However, actuated traffic signal timing only accommodates modest deviations from the nominal traffic conditions and is thus unable to respond to systematic changes in the traffic flow.

This paper proposes a *traffic predictive control* strategy for signalized intersections that predicts future traffic flow based on real-time measurements and adjusts the intersection’s signal timing accordingly. The prediction algorithm first identifies trends in historical traffic flow and then uses real-time measurements to determine the degree to which these historical trends are exhibited by the current traffic conditions. For example, historical trends may indicate that increased flow in one commute direction during the morning correlates with increased flow in the opposite direction in the evening. Traffic predictive control identifies this relationship and uses it to adjust signal timing parameters in the afternoon using measurements of traffic flow in the morning.

^{*}This research was supported by National Science Foundation SBIR Phase II Award 1329477

Email addresses: scoogan@ucla.edu (Samuel Coogan), chrisf@sensysnetworks.com (Christopher Flores), varaiya@berkeley.edu (Pravin Varaiya)

The present work thus lies within the extensive literature on traffic prediction and forecasting. See, for example, [5] for a survey of the literature. The majority of the forecasting literature focuses on freeways rather than urban street traffic [5]. For example, [6] and [7] develop a stochastic traffic flow model for freeways, and it is shown that this model is amenable to Kalman filtering techniques to estimate traffic conditions. Kalman filtering is also used in [8] to estimate traffic flow as the result of a random walk biased by historical increments in measured flow over time. In [9], k -means clustering is used to divide historical data, and an ARMAX prediction is computed for each possible cluster. The most likely cluster is used to provide a forecast of future traffic flow.

A large body of literature focuses on estimating flows throughout a network using flow measurements and conservation laws. For example, in [10], real-time link flow data is used to predict origin-destination and link flows by modeling a traffic network as a Gaussian Bayesian network, which provides conditional distributions and probability intervals for link flow. In [11], structural deviations from regular traffic patterns, modeled using polynomial trend filters, are used to estimate origin-destination flows. In contrast to these approaches, we focus on predicting traffic flow at a higher resolution for a single intersection.

A variety of model-based and statistical approaches exist for estimating travel times on arterial roads, which is closely associated with the problem of estimating traffic flows and volumes. A flow model is combined with GPS probe data in [12] to predict arterial travel times using a Bayesian network learning approach. Each link is modeled as being in a state of congestion or undersaturation, and the Bayesian network models the transition between these states. A similar approach is considered in [13] for using GPS probe data to estimate travel time where additional explanatory variables such as speed limits and number of lanes per link are used to reduce the number of parameters in the model. In [14], an approach for short-term travel time estimation that fuses past and real-time data is presented. These data are weighted based on their quality, which incorporates properties of the data including sensor accuracy and delay. A queuing theory approach is used to provide probability distributions on queue lengths in traffic networks in [15], and the model accounts for finite-capacity queues and captures spatial correlations. While the above approaches focus on aggregate estimates for larger networks, the focus of this paper is on high-resolution estimates of traffic flow for all movements at a single intersection.

In [16], aggregate daily traffic flow patterns are studied, and a functional principal component decomposition is used to reduce the dimensionality of the data. This approach is similar to the analysis proposed in Section 3 of the present paper, however, [16] focuses on identifying changes in daily traffic patterns for long term traffic monitoring, whereas, in the present paper, we use a principal component decomposition to lay the foundation for our traffic prediction approach. Moreover, [16] focuses on freeway networks and not signalized networks.

When prediction is applied to signalized intersections for adaptive control, prediction horizons are short, *e.g.*, seconds or minutes [17]. In contrast, the present work focuses on longer prediction horizons on the order of hours for use in conjunction with more traditional traffic signal timing methods such as pre-defined timing plans.

In this paper, we develop a principal-component based prediction scheme for signalized intersections using the *projection to latent structures (PLS)* algorithm [18]. Abstractly, this algorithm decomposes two sets of data to find low-rank, correlated structure between the data sets. For example, in the case of traffic, suppose the objective is to predict traffic flow from 2pm to 8pm using traffic flow measurements up to time 10am. First, historical measurements of traffic flow up to 10am and of traffic flow between 2pm and 8pm are collected. Next, the PLS algorithm decomposes the data into a set of pairs of *latent structures* which provide low-rank approximations of the data sets with the additional requirement that each pair of latent structures is highly correlated. Then, given real-time measurements of traffic flow for a particular day up to 10am, future traffic from 2pm to 8pm is predicted by computing weights for the latent structures.

An important property of this approach is that the proposed traffic predictive control builds on existing standard practices for traffic signal timing. In particular, we consider the common practice of signal timing based on time-of-day plans which are preprogrammed to apply during certain periods of the day [4, Chapter 5]. Each plan is designed to accommodate a certain level of traffic flow at the intersection. Traditionally, this level of traffic flow is determined based on averaged historical measurements; often, these measurements span only a limited time window over several days and may have been collected years ago. Critically,

averaged historical flow is unable to capture anomalous traffic patterns. In [19], a genetic algorithm selects from among a set of predefined plans based on current conditions.

The primary contribution of this paper is a traffic predictive control scheme that uses a prediction of future traffic flow to adjust the time periods for which the time-of-day plans are active and, additionally, suggests predicted levels of traffic around which the timing plans should be designed. Our approach thus does not depend on the exact algorithm that is used to determine timing plans (*i.e.*, green splits) from traffic flow, however, in our case study we employ a delay minimization policy. By ensuring that the proposed control approach is well-aligned with existing practices, the proposed controller integrates well with existing traffic control hardware which universally accommodate time-of-day timing plans and are often capable of remote changes to these plans. Additionally, practitioners familiar with standard practices are likely to be more receptive to the proposed traffic predictive control.

This paper is organized as follows: Section 2 describes the problem setup, available data, and details of the case study. Section 3 analyzes structural trends in the traffic flow data using a principal component analysis, which establishes the foundation for the prediction algorithm presented in Section 4. Section 5 proposes a traffic predictive control scheme that uses predictions of future traffic flow to adjust signal timing plans. Section 6 provides concluding remarks, future directions of research, and plans for implementation.

2. Preliminaries

We first characterize the requirements and assumptions generally and then specialize to a specific test site as our case study throughout the paper.

2.1. Available Data

We consider a single traffic intersection consisting of a set of M *turn movements* indexed 1 through M . For example, a prototypical intersection consists of four approaches, each approach consisting of a left turn movement, a right turn movement, and a through movement so that $M = 12$, as in the case below.

The flow rate of vehicles along each turn movement is measured and recorded once per interval of time Δ . Thus, $T \triangleq (24 \text{ hours})/\Delta$ measurements of the flow are made per movement per day. We assume measurements are available for a total of D days.

For day $d \in \{1, \dots, D\}$ and movement $m \in \{1, \dots, M\}$, let $x_m^d(t)$ denote the flow rate of vehicles executing the m -th turn movement on day d during time interval $t \in \{1, \dots, T\}$ in vehicles per hour (vph). From this notation, we aggregate the measurements into vectors and matrices as follows:

$$x_m^d = [x_m^d(1) \quad x_m^d(2) \quad \dots \quad x_m^d(T)]^\top, \quad d = 1, \dots, D, \quad m = 1, \dots, M \quad (1)$$

$$x^d = [(x_1^d)^\top \quad (x_2^d)^\top \quad \dots \quad (x_M^d)^\top] \quad d = 1, \dots, D \quad (2)$$

where $(\cdot)^\top$ denotes vector transpose. That is, $x_m^d \in \mathbb{R}^T$ is the vector of flow measurements along the m -th movement on day d and $x^d \in \mathbb{R}^{TM}$ is the vector of flow measurements along all M movements on day d .

We define the aggregated data measurement matrix as

$$X = \begin{bmatrix} (x^1)^\top \\ \vdots \\ (x^D)^\top \end{bmatrix} \in \mathbb{R}^{D \times (TM)}. \quad (3)$$

Typically, $TM \gg D$ so that X is a wide matrix. Table 1 contains a summary of the notation.

We consider measured turn movements to represent the exogenous demand on the system from the external environment. In particular, it is assumed that measured turn movements are negligibly affected by the choice of control action:

Assumption. *Turn movement flows originate from an exogenous process and, in particular, are not influenced by the choice of control actions at the intersection.*

| Notation | Meaning | Value (val) or Dimension (dim) for Case Study | |
|------------|--|---|-------|
| M | Number of turn movements at the intersection | 12 | (val) |
| Δ | Interval of time between subsequent measurement of turn movement flows | 15 min | (val) |
| D | Number of days | 132 | (val) |
| T | Number of measurements per movement per day | 96 | (val) |
| $x_m^d(t)$ | Flow rate of the m -th turn movement on day d at time t in vehicles per hour | 1 | (dim) |
| x_m^d | Vector of T measurements of the flow rate of the m -th turn movement on day d | 96 | (dim) |
| x^d | Vector of (TM) measurements of the flow rate on all movements on day d | 1152 | (dim) |
| X | Matrix of dimension $D \times (TM)$ containing measures of the flow rate on all movements for all days | 132×1152 | (dim) |

Table 1: Summary of notation.

For this assumption to be reasonable, we must have the sampling period Δ long enough so that the impact of the control signal is negligible (*e.g.*, if Δ is 5 minutes and the cycle time of the signal actuation at the intersection is 2 minutes, then a measurement $x_m^d(t)$ may include between two and three periods of actuation for movement m . Thus, Δ is too short since x_m^d will exhibit undesired oscillations caused by the control signal). Empirical evidence suggests that $\Delta \approx 15$ minutes is reasonable for minimizing such effects. Additionally, implicit in Assumption 2.1 is that vehicles do not reroute in response to changes in signal actuation. The test site presented below is a large intersection for which few alternative routes exist. Thus, the assumption that vehicles do not reroute is reasonable for our study.

2.2. Case Study Test Site

In this paper, we focus on a test site in Beaufort, South Carolina consisting of one intersection with four approaches. Each approach consists of a left turn movement, a right turn movement, and a through turn movement for a total of $M = 12$ movements. On the left of Figure 1 is a satellite image of the intersection and on the right of Figure 1 is a schematic depiction of the lane configuration and sensor placements at the intersection. A total of 44 magnetometer sensors provide real-time measurements of the movement of vehicles through the intersection; the sensors are manufactured by Sensys Networks, Inc. [20]. By measuring changes in magnetic field, the sensors are able to detect the presence of each vehicle at the intersection. Detection events from sensors on approaching lanes with subsequent detection events on departure lanes together with the signal phase are then used to determine the turn movement of each vehicle. These data are aggregated every $\Delta = 15$ minutes to provide a measurement of the number of vehicles executing each turn movement at the intersection. These data are reported in vehicles per hour. There are therefore a total of $T = 96$ measurements of flow per movement per day. This high-resolution data acquisition system provides a rich dataset for managing traffic at intersections [21].

The intersection consists of four *approaches* and four *departures*. A physically adjacent approach/departure pair is referred to as a *leg* of the intersection. As is labeled in Figure 1b, the left leg is the Eastbound (EB) leg, the right leg is the Westbound (WB) leg, the top leg is the Southbound (SB) leg, and the bottom leg is the Northbound leg (NB). Each approach consists of a left turn (LT) movement, a through (T) movement, and a right turn (RT) movement. We sometimes use, *e.g.*, the notation “ $m = \text{EB LT}$ ” to indicate the movement index corresponding to the Eastbound left turn movement.

Traffic at the case study intersection throughout the week has similar profiles each Monday through Thursday, and a different profile on Fridays, on Saturdays, and on Sundays. Figure 2 shows the average flow at the intersection for each of these groups for data spanning December 2014 to July 2015. Eleven

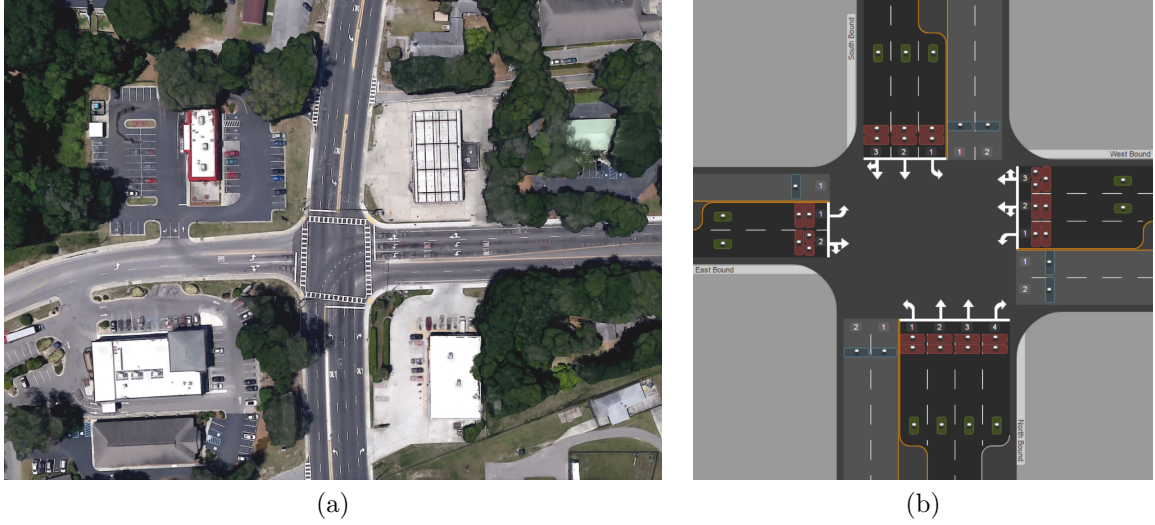


Figure 1: Test site in Beaufort, South Carolina. (a) Image of the test site which consists of one intersection with four approaches. Each approach consists of a left turn movement, a right turn movement, and a through turn movement. (b) Schematic depiction of the lane configuration and placement of sensors at the intersection. Stopbar sensors are indicated with red blocks, and departure lane sensors are indicated with blue blocks, and advance sensors placed upstream are indicated with green block. The sensors are manufactured by Sensys Networks, Inc. and their measurements are fused to determine the turn movement of each vehicle that transits the intersection.

days from this period are omitted due to missing measurements on these days. To ensure the figures are legible, the plots do not include traffic flows originating from or bound for the leftmost (that is, EB) leg (*i.e.*, movements WB T, NB LT, SB RT, and all EB movements) because these movements contribute much lower flow than the remaining movements (approximately 50–100 vph during peak periods).

The grouping depicted in Figure 2 is verified via standard k -means clustering for which, when sufficiently many clusters are computed, the clusters tend to contain days from only one of these four groups, but for more than four clusters, no meaningful division of the Monday–Thursday group is discernible.

3. Principal Components of Traffic Flow

In this section, we consider a low-rank decomposition of the measured traffic flow. This decomposition is obtained via a principal component (PC) analysis of the data, and we will see that this simple approach reveals much about the data. Furthermore, a PC analysis establishes the foundation for the PLS-based traffic prediction strategy that is the focus of Section 4 and is the main contribution of this paper.

3.1. Computation of Principal Components

Recall our data matrix X constructed in (3) from the vectors x^d , $d = 1, \dots, D$ containing the turn movement flow measurements for each movement over the course of each day d . Throughout the remainder of the paper, we focus exclusively on data from December 2014 to July 2015, Monday–Thursday for a total of $D = 132$ days. Define

$$\bar{x}_m = \frac{1}{D} \sum_{d=1}^D x_m^d \in \mathbb{R}^T \quad (4)$$

to be the mean measured flow along movement m over the course of a day, and define

$$\bar{x} = [\bar{x}_1^\top \quad \dots \quad \bar{x}_M^\top]^\top. \quad (5)$$

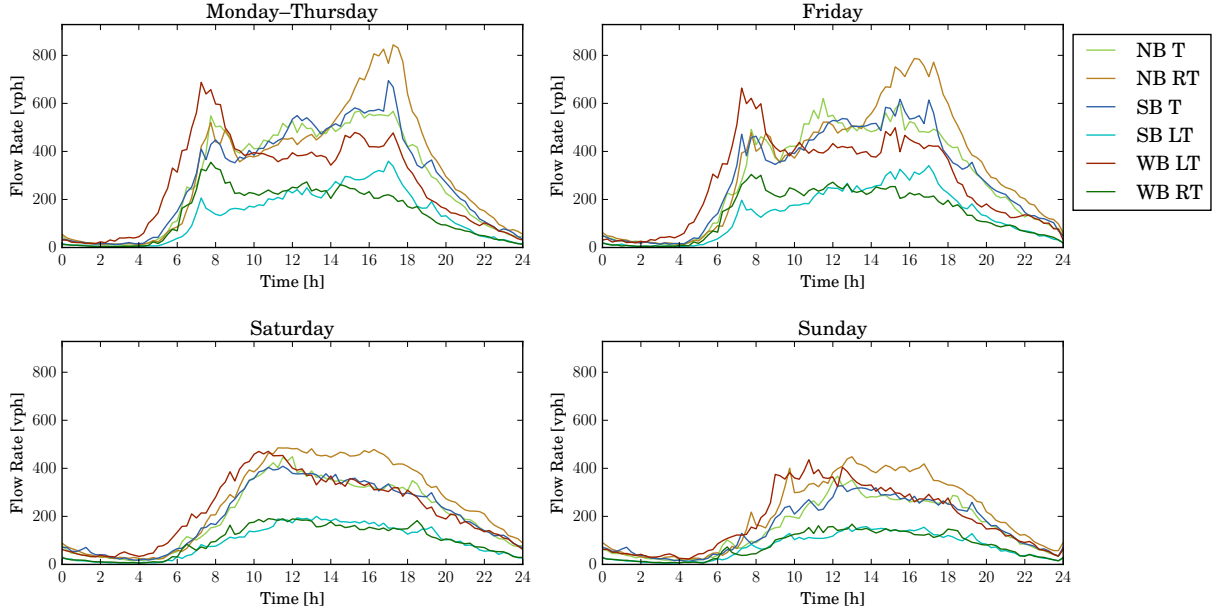


Figure 2: Average flow rates over a 7 month period from December 2014 to July 2015 for Monday–Thursday (132 days), Friday (32 days), Saturday (34 days), and Sunday (34 days). Traffic originating from and bound for the West (*i.e.*, EB LT, EB T, EB RT, NB LT, SB RT, and WB T movements) are not shown since traffic volumes for these movements are much lower than the remaining movements. Eleven days in this period contain missing measurements and are omitted from the analysis.

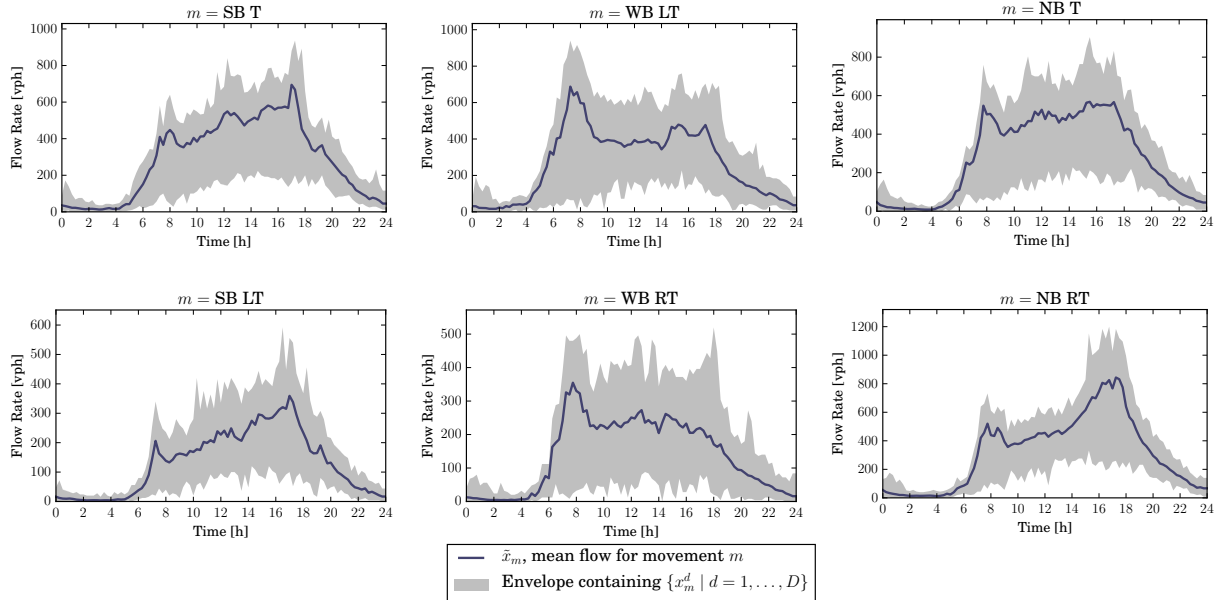


Figure 3: Plots of mean flow for six turn movements along with the envelope containing all measured data for all days Monday–Thursday.

The top-left image of Figure 2 plots \bar{x}_m for the six indicated movements. Figure 3 shows separately the mean measured flow for each of these six movements and additionally displays a shaded region that represents the envelope containing x_m^d for all $d = 1, \dots, D$. Clearly, there is a large variation in measured flow rate around the mean. Our first objective is to find a low-rank decomposition of the data to characterize this variation. Specifically, given the rank parameter $N \geq 1$, we wish to find a collection of N principal components q^1, q^2, \dots, q^N with each $q^i \in \mathbb{R}^{TM}$ and for each $d = 1, \dots, D$ a vector of weights $w(d) \in \mathbb{R}^N$ with

$$w(d) = [w^1(d) \quad w^2(d) \quad \dots \quad w^N(d)]^\top \quad (6)$$

such that

$$x^d \approx \bar{x} + \sum_{i=1}^N w^i(d) q^i, \quad (7)$$

that is, each of the 132 daily mean-centered measurement vectors $x^d - \bar{x}$ is approximately represented by a linear combination of the principal components. If (7) holds, then we may effectively replace x^d by its weight vector $w(d)$. As will be evident below, much of the day-to-day variation can be captured by a few (three to five) principal components.

To make our search for principal components precise, let $\tilde{x}^d = x^d - \bar{x}$ for all $d = 1, \dots, D$ and define

$$\tilde{X} = \begin{bmatrix} (\tilde{x}^1)^\top \\ \vdots \\ (\tilde{x}^D)^\top \end{bmatrix} = X - \mathbf{1}_D \bar{x}^\top \quad (8)$$

where $\mathbf{1}_n$ denotes the all-ones vector of length n , and let

$$Q = [q^1 \quad \dots \quad q^N] \in \mathbb{R}^{(TM) \times N} \quad (9)$$

$$W = \begin{bmatrix} w^\top(1) \\ \vdots \\ w^\top(D) \end{bmatrix} \in \mathbb{R}^{D \times N} \quad (10)$$

where $w^\top(d)$ denotes the transpose of the weight vector $w(d)$.

We reformulate (7) and specifically seek Q and W to minimize

$$\|\tilde{X} - WQ^\top\|_F \quad (11)$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. It is well known that a pair (W, Q) minimizing (11) is obtained via the singular value decomposition of \tilde{X} . This decomposition results in q^1, \dots, q^N that are orthonormal and are assumed ordered with respect to the (descending) order of the singular values of \tilde{X} . In this way, q^1 is the principal component that lies in the direction maximizing the explained variance of the data, *i.e.*, maximizes $\sum_{d=1}^D (q^1)^\top \tilde{x}^d$, q^2 is the principal component that lies in the direction maximizing the explained variance subject to the constraint that $(q^1)^\top q^2 = 0$, *etc.* It is standard to assume without loss of generality that each q^i is of unit norm; here, we multiply each q^i by a factor of 100 and therefore divide each $w^i(d)$ by a factor of 100 to make the plots more intuitive and legible.

3.2. Case Study

In Figure 4, we plot the first four principal components of our dataset. Each q^i is TM -dimensional, however, for ease of comprehension, each principal component is plotted as M traces of length T , each trace corresponding to one turn movement. From the figure, we observe identifiable trends in each of the principal components, which are discussed next. We also note that the principal components exhibit some high-frequency oscillation, especially in the AM peak period. This oscillation is generally of lower magnitude than the main trends in the components and would likely be ameliorated with more data.

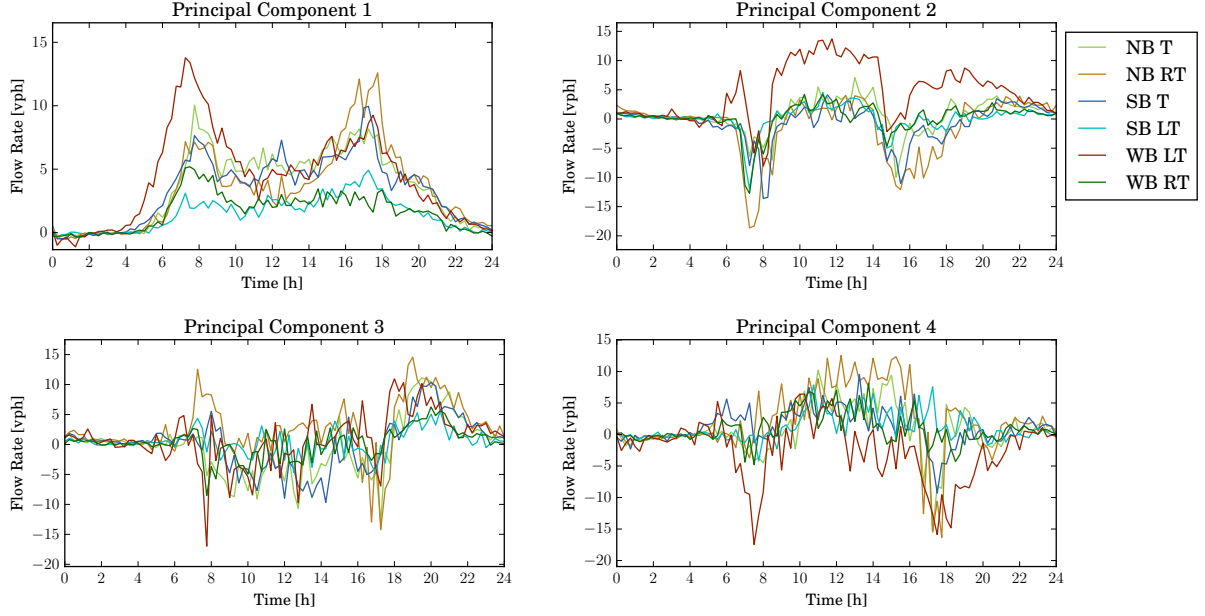


Figure 4: The first four principal components of the traffic data. Each component is TM -dimensional but is plotted as M traces of length T in the figures, each trace corresponding to one turn movement over 24 hours.

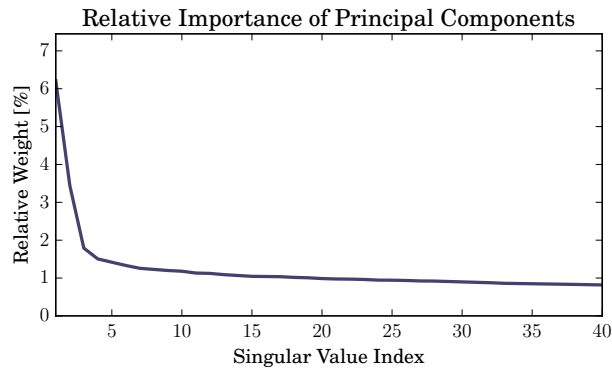


Figure 5: Relative weight of the singular values obtained from a singular value decomposition of \tilde{X} . The singular values are ordered by magnitude and the relative weight is a percent of the sum of all singular values. The plot indicates that a significant portion of the variation in traffic flow is explained by the first few principal components of the decomposition.

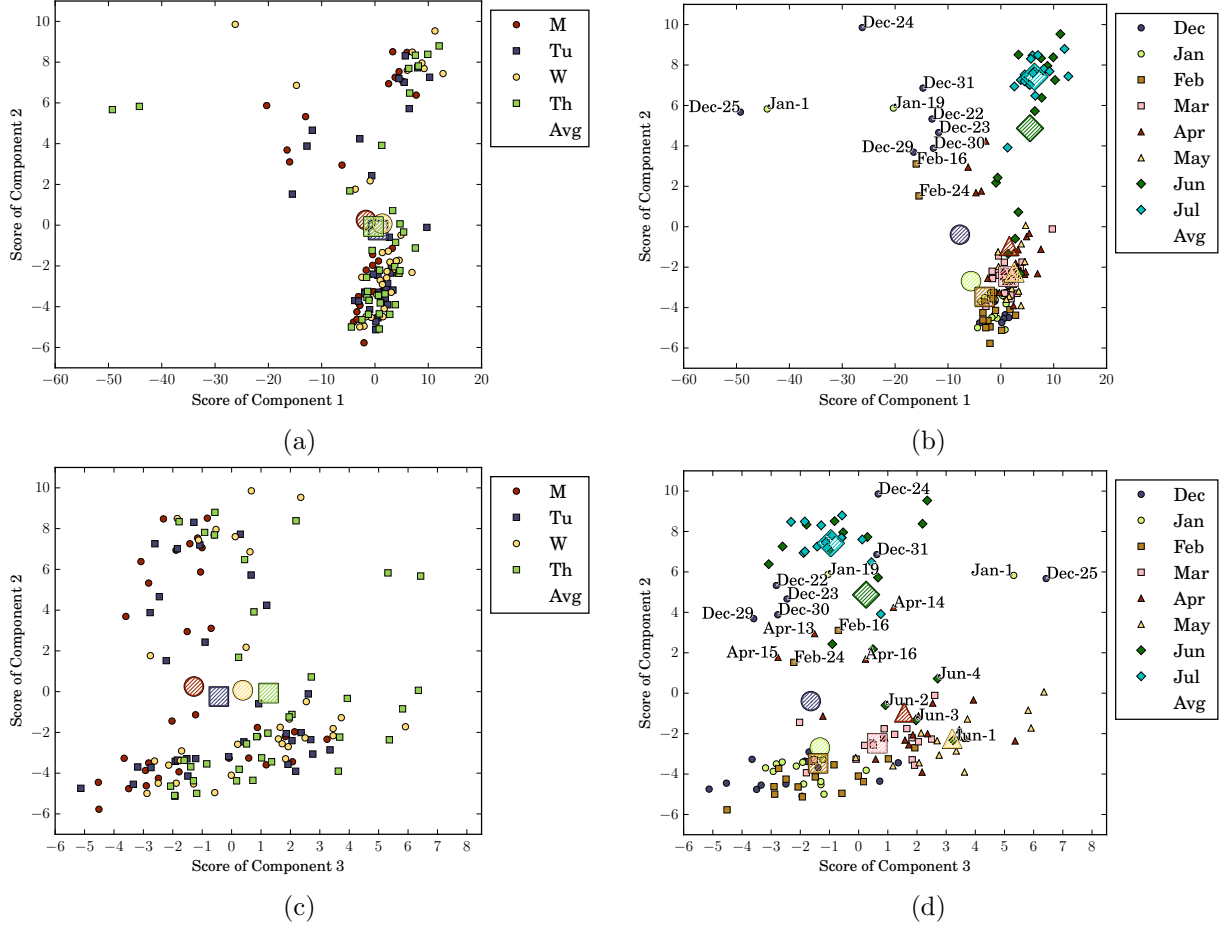


Figure 6: Scatter plots of weights for pairs of principal components. Each day is indicated by a small marker and the centroid average for each label category is indicated with a large hatched marker. (a) The weight of component 1 versus the weight of component 2 for each day, labeled by day of week. The centroid points for all four days are close, supporting the notion that no further clustering by day-of-week is necessary. (b) The weight of component 1 versus the weight of component 2, labeled by month. A large negative weight for component 1 indicates days when work/business traffic is low. (c) The weight of component 2 versus the weight of component 3 labeled by day. There is a relationship between day of week and the weight of component 3 but it is small compared to the spread of weights. (d) The weight of component 2 versus the weight of component 3 labeled by month. Positive weights for component 3 indicate days when school is not in session.

Figure 5 plots the singular values of \tilde{X} normalized by the sum of all singular values. From the plot, we see that a significant portion of the variation in the traffic flow is explained by the first few principal components of the decomposition.

In Figure 6, we present scatter plots of the weights for pairs of components. In Figure 6a we plot the weight of component 1 versus the weight of component 2 for each day where the data is labeled by the day of the week. In Figure 6b we again plot the weight of component 1 versus the weight of component 2 but now label the data by month. Similarly, Figure 6c and Figure 6d plot the weight of component 3 versus the weight of component 2 and label the data by day and by month, respectively. Each marker corresponds to one day and the centroid average for each label category is indicated with a large hatched marker.

We now discuss some clear trends that emerge from these plots. In Figure 6b, it is apparent that a large negative weight for component 1 indicates a nonbusiness day; all of the labeled days are on or near holidays in the U.S. except February 24 for which there were school and business closings due to weather conditions, *i.e.*, it was a snow day. This observation is congruous with the plot of principal component 1 in Figure 4 which corresponds to overall higher traffic volume, especially during the morning and afternoon periods. Thus, a negative score for component 1 indicates lower traffic during these periods. Furthermore, the morning and afternoon peaks for component 1 in Figure 4 correspond to reversed commute directions, *i.e.*, the morning peak for WB LT corresponds with the afternoon peak of NB RT, likewise for the NB T peak in the morning and the SB T peak in the afternoon.

Approximately one mile to the north of the intersection is a school. In Figure 6d, we see that a positive score for component 2 indicates days when school is not in session, which includes the entire months of June and July except June 1–4, which are labeled. We additionally label dates when school is not in session due to holidays or weather conditions; April 13–16 is the Spring Break holiday. Again, this observation is congruous with the plot of principal component 2 in Figure 4, which exhibits two particularly telling features. First, this component has narrow spikes around 7:30 and 15:30, corresponding to the school session. Second, the WB LT movement clearly differs from the others; this is the only movement that is not going to or coming from the north (*i.e.*, Southbound leg) where the school is located.

Figure 6c indicates that the weight of component 3 generally increases from Monday to Thursday, although the increase is modest compared to the spread of the weights for this component, and the centroids in Figure 6a are closely clustered. Thus these plots generally support our decision to cluster Monday–Thursday together.

Lastly, we observe longer term trends in the data. In Figure 6d, the score of component three generally increases from December to May, the months when school is in session. This may correspond with seasonal variations; as the plot of principal component 3 in Figure 4 suggests, a larger score for component 3 indicates more traffic in the evening. The days get longer from December to May, which may explain higher evening traffic. In Figure 6b, there is also an increase in the score of component 1 per month for all months December to July, although this trend is somewhat obscured by the axis scaling. This suggests that for the entire data set, there is a general increase in business-related traffic, which may be seasonal or may reflect improving economic conditions from December 2014 to July 2015.

3.3. Discussion

By its nature, a PC-based approach offers a more nuanced interpretation of the data than cluster-based approaches, as we saw above, since it indicates the *degree* to which a set of measurements exhibits a particular component. For example, rather than simply identifying that traffic patterns are different in the Winter season than in the Spring season (which clustering may reveal), we are able to quantify this difference. Furthermore, a PC-based approach retains the ability to identify categorical differences in the data. For example, our analysis indicates that traffic patterns for days when school is in session are qualitatively different than when school is not in session.

We also note that in a PC-based approach the components frequently offer an interpretation or explanation for the observed trends. For example, we could associate each of the observed trends in the scatter plots of Figure 6 with an intuitive hypothesis regarding the variation in traffic movement flows based on the principal components in Figure 4. We note that the trends observed above are unique for the available dataset. This means that the principal components and corresponding interpretations are unlikely to

transfer to other datasets obtained from other traffic intersections or networks. An interesting direction for future research is to study the transferability of prediction and learning algorithms to other datasets.

4. Traffic Prediction from Low-Rank Structure

Suppose it were possible to obtain an estimate of the weight vector $w(d)$ for some day $d \in D$ by using measurements only up to some time $T^* < T$, that is, from the vector

$$z^d = [(z_1^d)^\top \quad \dots \quad (z_M^d)^\top]^\top \in \mathbb{R}^{T^*M} \quad (12)$$

where

$$z_m^d = [x_m^d(1) \quad \dots \quad x_m^d(T^*)]^\top \in \mathbb{R}^{T^*} \quad m = 1, \dots, M. \quad (13)$$

Then it would be possible to *predict* traffic flow for times $t > T^*$ by constructing an estimate of x^d as $\hat{x}^d = Q\hat{w}(d)$ (see (7)) where \hat{x}^d is our estimate of the traffic flow for the entire day and $\hat{w}(d)$ is our estimate of the weight vector $w(d)$.

A naive approach to estimating $w(d)$ using measurements up to time T^* is to project the vector z^d onto the corresponding truncation of each q^i . However, this approach is unreliable because the PC-based decomposition uses data for the whole day and does not consider that we wish to make a prediction after time T^* . For example, consider principal component 3 in Figure 4 which, as argued in Section 3.2, corresponds with increased/decreased traffic in the evening. This component lies largely in the direction corresponding to flow measurements later in the day and thus it is difficult to predict $w^3(d)$ given z^d for T^* early in the day.

To overcome this limitation, we propose computing a set of components that simultaneously explains the variation in the measured data *before* time T^* and correlates with a second set of components that explains the variation in the traffic flow data measured *after* time T^* . To make this precise, we define the following vector of flow measurements after time T^* , complementary to z^d defined in (12)–(13):

$$y^d = [(y_1^d)^\top \quad \dots \quad (y_M^d)^\top]^\top \in \mathbb{R}^{(T-T^*)M} \quad (14)$$

where

$$y_m^d = [x_m^d(T^* + 1) \quad \dots \quad x_m^d(T)]^\top \in \mathbb{R}^{(T-T^*)}, \quad m = 1, \dots, M. \quad (15)$$

Let

$$\bar{z} = \frac{1}{D} \sum_{d=1}^D z^d, \quad \bar{y} = \frac{1}{D} \sum_{d=1}^D y^d \quad (16)$$

and define

$$Z = \begin{bmatrix} (z^1)^\top \\ \vdots \\ (z^D)^\top \end{bmatrix}, \quad Y = \begin{bmatrix} (y^1)^\top \\ \vdots \\ (y^D)^\top \end{bmatrix} \quad (17)$$

$$\tilde{Z} = Z - \mathbf{1}_D \bar{z}^\top, \quad \tilde{Y} = Y - \mathbf{1}_D \bar{y}^\top. \quad (18)$$

The matrix Z collects all flow measurements up to time T^* and the matrix Y collects all flow measurements after time T^* onward to the final time T . The matrices \tilde{Z} and \tilde{Y} are mean-centered versions of Z and Y . Note that Z and Y partition X , and, similarly, \tilde{Z} and \tilde{Y} partition \tilde{X} .

Our objective is to find a collection of *predictor* components p^1, p^2, \dots, p^N with each $p^i \in \mathbb{R}^{T^*M}$; a collection of *predicted* components c^1, c^2, \dots, c^N with each $c^i \in \mathbb{R}^{(T-T^*)M}$; and, for each $d = 1, \dots, D$, a vector of weights $\omega(d) \in \mathbb{R}^N$ with

$$\omega(d) = [\omega^1(d) \quad \omega^2(d) \quad \dots \quad \omega^N(d)]^\top \quad (19)$$

such that

$$z^d \approx \bar{z} + \sum_{i=1}^N \omega^i(d) p^i \quad (20)$$

$$y^d \approx \bar{y} + \sum_{i=1}^N \omega^i(d) c^i. \quad (21)$$

Let

$$\omega^i = [\omega^i(1) \quad \omega^i(2) \quad \dots \quad \omega^i(D)]^\top \in \mathbb{R}^D \quad \text{for all } i = 1, \dots, N. \quad (22)$$

The use of (20)–(21) for prediction is immediately apparent: if we are able to determine the weights $\omega(d)$ using only z^d , that is, measurements up to time T^* , then we are able to predict traffic flow after time T^* using the same weights and the collection of predicted components c^i , $i = 1, \dots, N$. We formalize this prediction procedure in Section 4.2.

4.1. The Projection to Latent Structures Algorithm

To compute the collection of predictor and predicted components, we use a statistical technique called *projection to latent structures (PLS)*, also known as *partial least squares*. Given two sets of measured variables (*e.g.*, measured traffic flow over two time intervals) Z and Y , the PLS algorithm identifies low-rank approximations of both sets as in (20)–(21) in such a way that the low-rank components are highly correlated. The low-rank approximations are then used for prediction. While the PLS algorithm is a general statistical tool, it has particularly been developed in the domain of chemometrics beginning with the work of Wold *et al.* [22] and has since been applied to a wide range of chemical data analysis problems. References [23], [24], and [25] provide early overviews of the PLS algorithm with emphasis on applications to chemometrics, and references [18] and [26] provide more recent tutorials on the PLS algorithm.

The PLS technique is iterative; it first computes a pair of components (p^1, c^1) from the data matrices \tilde{Z} and \tilde{Y} , then the algorithm deflates the data matrices by removing the contribution of this pair of components. Next, a new pair (p^2, c^2) is computed from the updated data matrices, *etc.* To determine the first pair (p^1, c^1) , we solve the following optimization problem:

$$(r^*, s^*) = \arg \max_{r, s} (\tilde{Z}r)^\top (\tilde{Y}s) \quad (23)$$

$$\text{such that} \quad \|r\|_2 = 1 \quad (24)$$

$$\|s\|_2 = 1. \quad (25)$$

The interpretation of (23) is as follows: we wish to find directions $r^* \in \mathbb{R}^{T^*M}$ and $s^* \in \mathbb{R}^{(T-T^*)M}$ to maximize the empirical covariance of the score vectors $u = \tilde{Z}r^* \in \mathbb{R}^D$ and $v = \tilde{Y}s^* \in \mathbb{R}^D$. These score vectors contain the projection of each day's data z^d and y^d onto the directions r^* and s^* . We define the first score component vector

$$\omega^1 \triangleq \frac{1}{\|\tilde{Z}r^*\|_2} \tilde{Z}r^*. \quad (26)$$

Note that r^* and s^* are the first left and right, respectively, singular vectors of $\tilde{Z}^\top \tilde{Y}$. To obtain p^1 we project \tilde{Z} onto ω^1 , and, similarly, to obtain c^1 we project \tilde{Y} onto ω^1 :

$$p^1 = \tilde{Z}^\top \omega^1 \quad (27)$$

$$c^1 = \tilde{Y}^\top \omega^1. \quad (28)$$

Note that our treatment of the data matrices \tilde{Z} and \tilde{Y} is asymmetric, that is, ω^1 is obtained from \tilde{Z} and r^* . This asymmetry is because we ultimately wish to use the scores and components for *prediction*. We next deflate the data matrices \tilde{Z} and \tilde{Y} using the first score vector ω^1 and the computed components:

$$\tilde{Z}^+ = \tilde{Z} - \omega^1 p^1 \quad (29)$$

$$\tilde{Y}^+ = \tilde{Y} - \omega^1 p^1. \quad (30)$$

Above, \tilde{Z}^+ and \tilde{Y}^+ are the updated data matrices. To compute ω^2 , p^2 , and c^2 we repeat the above procedure, replacing \tilde{Z} and \tilde{Y} with their updated versions \tilde{Z}^+ and \tilde{Y}^+ . We repeat this process until we obtain N score vectors and components where N is a design parameter. We gather the computed scores and components:

$$\Omega = [\omega^1 \quad \dots \quad \omega^N] \in \mathbb{R}^{D \times N} \quad (31)$$

$$P = [p^1 \quad \dots \quad p^N] \in \mathbb{R}^{(T^*M) \times N} \quad (32)$$

$$C = [c^1 \quad \dots \quad c^N] \in \mathbb{R}^{((T-T^*)M) \times N}. \quad (33)$$

We then have $\tilde{Z} \approx \Omega P^\top$ and $\tilde{Y} \approx \Omega C^\top$.

Above, to compute r^* and s^* , one approach is to first multiply \tilde{Z}^\top and \tilde{Y} and then compute the singular value decomposition of the product, and this operation is performed for each iteration. For traffic flow measurements, it is typically the case that $D \ll \min\{T^*M, (T-T^*)M\}$, thus $\tilde{Z}^\top \tilde{Y}$ is a large matrix of dimension $T^*M \times (T-T^*)M$ and computing its singular value decomposition is computationally expensive. There exists efficient implementations of the PLS algorithm that use the *kernel* matrices $\tilde{Z}\tilde{Z}^\top$ and $\tilde{Y}\tilde{Y}^\top$ to compute the predictor and prediction components by finding the principal eigenvector and eigenvalue pair of a (much smaller) $D \times D$ matrix instead [27]. The PLS algorithm is described by (23)–(33).

4.2. Prediction From Latent Structures

Once the predictor and predicted components have been computed, we are able to construction a prediction of future traffic flow given sample measurements. To this end, let

$$z^s \in \mathbb{R}^{T^*M} \quad (34)$$

denote measured traffic flow for the M turn movements up to time T^* . Our objective is to use z^s along with the predictor and predicted components calculated from historical data to predict traffic flow for our sample day for time periods after T^* . To do so, we first calculate $\hat{\omega}$, a vector of scores for the sample day. As above, let \bar{z} and \bar{y} denote means of historical traffic flow up to time T^* and after T^* , respectively. The score vector is computed as

$$\hat{\omega} = ((z^s - \bar{z})^\top (P^\top)^\dagger)^\top \in \mathbb{R}^N \quad (35)$$

where $(P^\top)^\dagger$ denotes the Moore-Penrose pseudoinverse of P^\top . Then

$$\hat{y}^s = \hat{\omega}^\top C^\top + \bar{y} \in \mathbb{R}^{(T-T^*)M} \quad (36)$$

is the prediction of traffic flow after time T^* . Combining (35) and (36), we obtain one succinct equation for prediction:

$$\hat{y}^s = (z^s - \bar{z})^\top (P^\top)^\dagger C^\top + \bar{y}. \quad (37)$$

Notice that after P^\top and C^\top are computed via the PLS algorithm and the pseudoinverse of P^\top is computed and stored, predicting traffic flow given a sample vector of measurements requires only matrix multiply operations and is thus computationally easy.

4.3. Extensions and Case Study

We now discuss a few immediate extensions to the PLS prediction algorithm presented above that we utilize in Section 5. Firstly, we assume above that our objective is to predict flow measurements for all time periods beyond time T^* up to the end of the day, T . However, our objective may be to predict traffic flow for a shorter horizon or for a period not immediately subsequent to the measurement period. For example, we may wish to predict traffic during the evening from observed traffic flow in the morning. We can accommodate this scenario by appropriately truncating the matrices Z and Y above.

Secondly, we assume above that the interval of time between subsequent flow measurements, Δ , is the same for the measured data as well as the prediction. However, it is straightforward to accommodate differences in these intervals, or even nonuniform intervals so long as the measurement times are the same for each day. For example, we may wish to use flow measurements at 15 minutes intervals up to time T^* to predict traffic flow at one hour intervals subsequently.

To demonstrate the PLS prediction algorithm as well as these extensions, we return to our case study. We let T^* correspond to 10:00 and consider the case when flow measurements are available up to T^* in 15-minute increments and available thereafter in 1-hour increments. Figure 7 shows the first three predictor and predicted components generated from the dataset. To emphasize the different measurement intervals, individual data points are indicated with a solid marker point. Comparing Figure 7 to Figure 4, we see that each predictor/predicted component pair of Figure 7 resembles the corresponding principal component of Figure 4. The differences reflect that the predictor and predicted components are calculated to maximize their correlation.

Figure 8 illustrates the use of the PLS algorithm for predicting traffic flow. We consider three unusual days from our data set: February 24, which, as described above, experienced winter weather resulting in school closures; July 2, which is a Thursday and preceded a long holiday weekend in the U.S.; and January 1, which is a significant holiday and resulted in dramatically different traffic patterns. As a demonstration, we only plot results for two of the twelve movements; the first (respectively, second) row of Figure 8 plots flow measurements and predictions for the SB through movement (respectively, NB right turn movement).

The blue trace shows the average flow over the fourteen one-hour periods from 10:00 to midnight, the gold trace shows the actual flow on the given days, and the green trace shows the flow as predicted using (37) with four predictor/predicted component pairs (three of which are shown in Figure 7). To compute the prediction, we employ leave-one-out cross validation whereby, for each unusual day, the predictor/predicted components are computed from the dataset excluding the sample day of interest. From the plots, we see that the PLS algorithm correctly predicts the below-average flow on February 24 and the above-average flow on July 2 where the algorithm nearly exactly predicts the peak flow for both movements between 17:00 and 18:00. In addition, the algorithm adeptly predicts the well below-average flow on January 1 for which traffic conditions greatly differ from the norm. Thus, while the PLS algorithm is ultimately a linear prediction scheme as is apparent in (37), the technique ably accommodates significant variation in the traffic flow.

To quantitatively assess the quality of the prediction, we compute a prediction for each of 132 days in the dataset using leave-one-out cross validation and compute the one-norm distance from the prediction to the actual flow measurements. That is, for each day $d = 1, \dots, D$, we compute the prediction error E_{pred}^d as

$$E_{\text{pred}}^d = \|y^d - \hat{y}^d\|_1 = \sum_{m=1}^M \sum_{t=T^*+1}^T |y_m^d(t) - \hat{y}_m^d(t)| \quad (38)$$

where \hat{y}^d is the predicted traffic flow on day d computed as in Section 4.2. We use the one-norm as our distance metric because it corresponds to the absolute total difference of the number of vehicles that are measured versus predicted at the intersection along each movement. Likewise, we define the baseline error to be the difference between the average flow measurements and the actual flow measurements¹:

$$E_{\text{base}}^d = \|y^d - \bar{y}\|_1. \quad (39)$$

¹Since we employ a cross validation scheme, \bar{y} will change slightly for each day because we remove a different set of measurements from the dataset in each case.

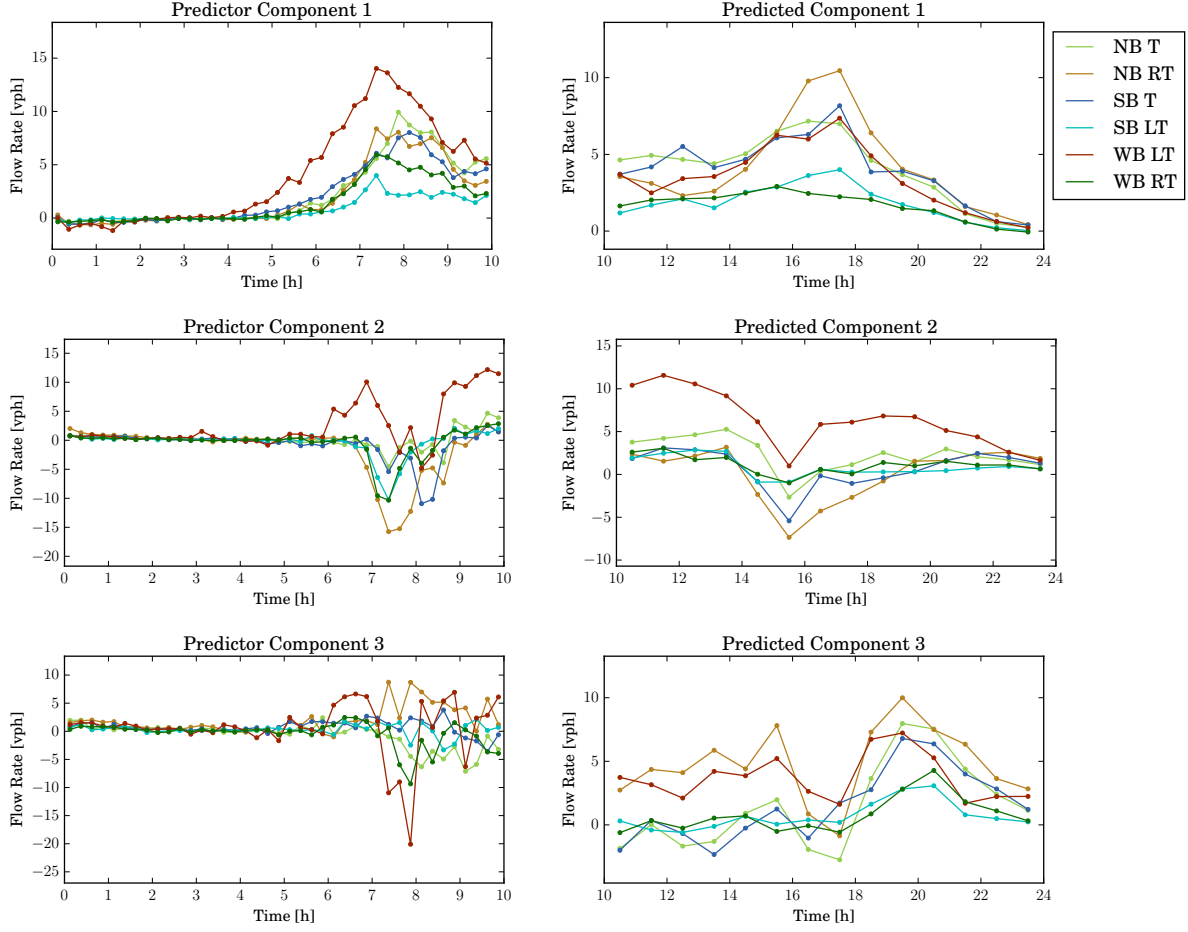


Figure 7: Projection to latent structures algorithm. We take T^* corresponding to 10:00 and consider the case for which flow measurements are available in 15-minute increments up to 10:00 and available in 1-hour increments thereafter. The left column of plots contains the first three predictor components, the right column of plots contains the corresponding prediction components.

Figure 9 is a scatter plot of the normalized error decrease $(E_{\text{base}}^d - E_{\text{pred}}^d)/E_{\text{base}}^d$ versus the baseline error with no prediction E_{base}^d , for each day $d = 1, \dots, D$. Excluded are December 24 and January 1 which both have baseline errors exceeding 14 000 with prediction errors less than 5 500, an error decrease of over 60%. Nearly all of the points (113 out of 132) correspond to positive error decreases indicating that the prediction almost always is an improvement over the baseline average. The circles (respectively, crosses) are those days for which the total flow volume throughout the day is below (respectively, above) average. We see that the PLS algorithm performs well in both cases. For most days, the improvement in error is approximately 10% to 20%. However, for those days for which the baseline error is high, the improvement is higher and can be as high as 50% to 60%, and this holds for both those days for which traffic is below average and those days for which traffic is above average. Thus for days that differ significantly from the average, the PLS prediction is especially accurate.

There is one particularly anomalous point for which the baseline error is high, approximately 7 000, but the normalized error decrease is not significant and is in fact slightly negative. It appears that, on this day, total traffic volume was only moderately higher than normal but traffic flow was significantly above average for some movements and below average for other movements, although the cause is unclear. While the PLS algorithm was unable to predict this difference, we observe that the PLS algorithm did not result

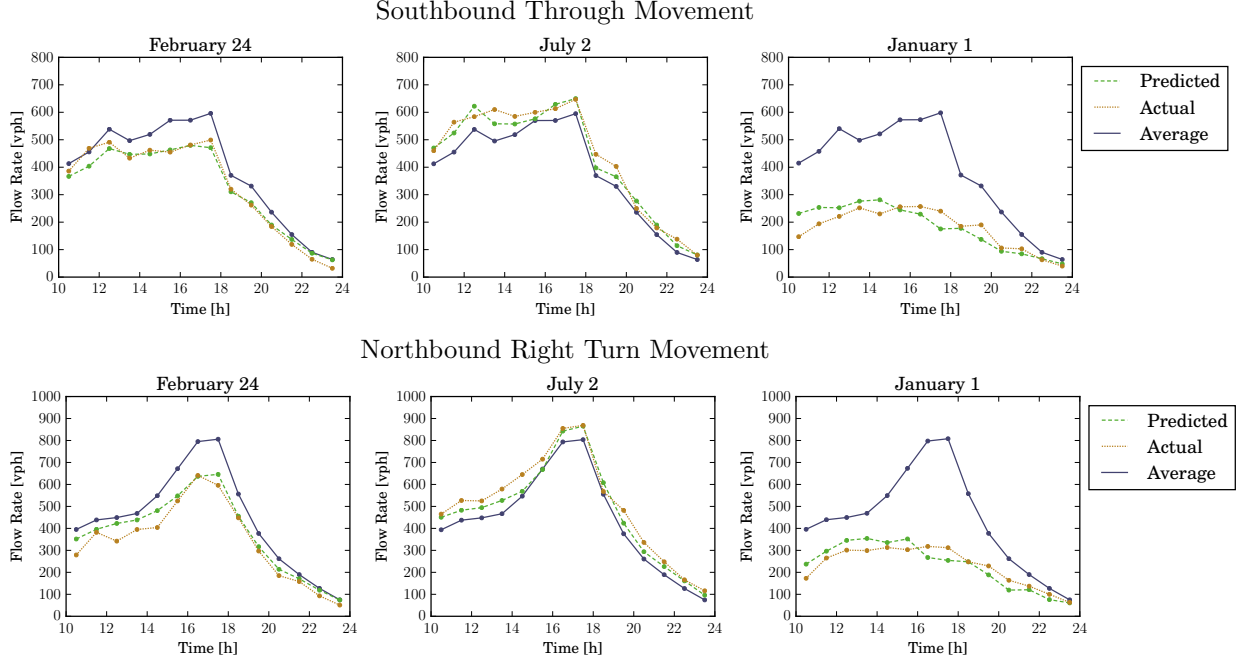


Figure 8: Example prediction results on three unusual days for two movements. On February 24, the algorithm correctly predicts below average flow caused by winter weather and closed schools. On July 2, the algorithm correctly predicts above average flow caused by the subsequent holiday weekend. The algorithm additionally predicts the well below average traffic on the holiday January 1 which deviates substantially from the average flow.

in a prediction error that was significantly worse than the baseline error.

5. Traffic Predictive Control

How do we utilize traffic flow predictions based on low-rank structure to improve traffic control? We first review a common technique for programming traffic signal controllers and describe an approach for optimizing this technique using only the average flow measurements. We will then use traffic predictions from the PLS algorithm to improve upon this approach. A primary goal is to ensure the algorithm is easily applied to existing hardware and well-aligned with conventional traffic control practices.

5.1. Time-of-Day Signal Operation

A common approach to traffic control supported by nearly all traffic control hardware is *time-of-day (TOD)* scheduling [4, Chapter 5] whereby a pre-defined control plan is applied during specified periods of the day. For example, a pre-defined plan may be specified for 6:00 to 9:00, and a different plan may be specified for 9:00 to 13:00, *etc.* Typically, the TOD periods are obtained via limited data collection at the intersection or simply by prior experience.

Once the TOD periods have been selected, timing parameters are designed for each period and applied to the controller, constituting a TOD plan. Typical timing parameters include the cycle time of the intersection and green time allocations for the turn movements. A TOD plan may be completely fixed or it may be *actuated* in which case the presence or absence of queued vehicles extends green time allocations by a certain amount or leads to early termination of a phase actuation.

The timing parameters for a given TOD period are determined by expected turn movement flows at the intersection. Since the TOD plan is applied at fixed periods and the parameters are fixed, the plan is designed around a set of nominal turn movement flows, for example, average turn movement flows over

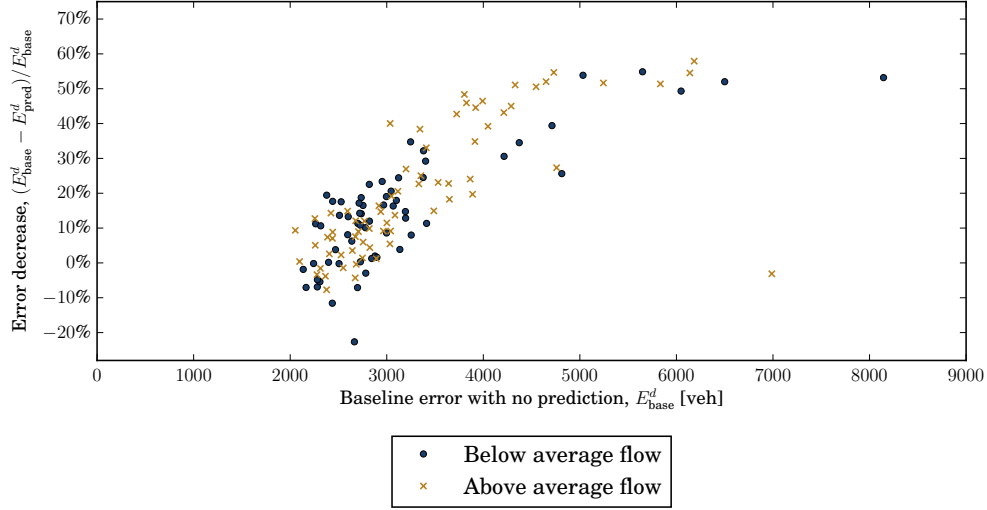


Figure 9: Scatter plot of prediction error. The plot shows baseline error E_{base}^d versus the normalized error decrease $(E_{\text{base}}^d - E_{\text{pred}}^d)/E_{\text{base}}^d$ for each day $d = 1, \dots, D$. Days with below average (respectively, above average) flow are indicated with circles (respectively, crosses). For clarity, the plot excludes December 25 and January 1 which experience error decreases exceeding 60% with baseline errors exceeding 14000.

the TOD period. Methods for determining desirable timing parameters given nominal turn movement flows have a long history going back at least to the seminal work of [28] and is not the focus of this paper. Instead, we assume that what is required for signal timing is only an estimation of the turn movement flows during a TOD period, for which any signal timing optimization scheme may be employed. This approach is reasonable if traffic flow does not deviate excessively from these estimated flows during a TOD period. In the following section, we suggest an algorithm for identifying TOD periods that minimize such deviations.

5.2. Optimal Time-of-Day Segmentation

We focus on the problem of determining TOD periods and representative turn movement flows for each period. In the sequel, we will consider adjusting the TOD periods and representative turn movement flows based on predictions using the PLS algorithm.

To this end, we now present an algorithm for optimally segmenting the 24-hour day into TOD periods. The intuitive idea is to identify segmentation times that minimize the variability in turn movement flows within each TOD period so that a particular fixed TOD plan works well throughout the period. For example, turn movement flows may remain relatively steady through the morning period but change substantially in the afternoon, requiring a different TOD plan. Increasing the number of TOD periods reduces variability in any given period, but there are practical limitations to the number of TOD periods that may be employed at an intersection. For example, it typically takes several cycles totaling up to ten minutes to fully switch from one TOD plan to another; during this intermediate time, mobility at the intersection may be reduced. Furthermore, excessive changes to timing plans may confuse drivers. Traffic intersections commonly employ up to seven TOD plans throughout the day.

Approaches for TOD segmentation proposed in the literature include randomized clustering algorithms [29, 30], heuristic genetic algorithms [31], and simulation based algorithms [32]. To improve performance, the paper [30] suggests explicitly incorporating time as a variable in determining the clusters. Here, we consider an optimal segmentation approach suggested for generic data sets in [33] and adapt it to the context of traffic flow measurements. Like [30], this approach explicitly accounts for time when identifying clusters. However, our approach is not based on k -means clustering and is able to identify optimal clusters

in computational time quadratic in the number of time steps. In contrast, k -means clustering requires exponential computational time, although efficient heuristics exist.

Suppose we wish to obtain S TOD periods for some $S > 1$. We consider the equivalent problem of choosing $S - 1$ segmentation times $\tau_1, \tau_2, \dots, \tau_{S-1}$ with each $\tau_i \in \{1, 2, \dots, T - 1\}$ satisfying

$$\tau_1 < \tau_2 < \dots < \tau_{S-1} \quad (40)$$

so that

$$\{\tau_{i-1} + 1, \tau_{i-1} + 2, \dots, \tau_i\} \quad (41)$$

defines the i th TOD period for $i \in \{1, \dots, S\}$, with $\tau_0 := 0$ and $\tau_S := T$.

To assess the quality of a given segmentation $(\tau_1, \dots, \tau_{S-1})$, we consider a vector of turn movement flows $x \in \mathbb{R}^{TM}$ throughout the day and define the *cost* of a time segment $(t_a, t_a + 1, \dots, t_b)$ given these flows for some $t_a \leq t_b$ as follows:

$$\text{Cost} = \min_{\mu \in \mathbb{R}^P} F(t_a, t_b, x, \mu) \quad (42)$$

where $\mu \in \mathbb{R}^P$ is a parameter vector of dimension $P \geq 1$ and $F(t_a, t_b, x, \mu)$ is a positive function that is convex in μ . We call $F(t_a, t_b, x, \mu)$ the *fit* of x with the parameter vector μ on time segment $(t_a, t_a + 1, \dots, t_b)$. In this way, $F(\tau_{i-1} + 1, \tau_i, x, \mu)$ is the fit of x with μ on TOD period i . For $t_b < t_a$, we define the fit F to be 0. To make this concrete, in this paper, we assume $P = M$ so that $\mu \in \mathbb{R}^M$ and take

$$F(t_a, t_b, x, \mu) = \sum_{t=t_a}^{t_b} \Phi(x(t), \mu) \quad (43)$$

for $\Phi : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}_{\geq 0}^M$ a positive function that is convex in its second argument where $\mathbb{R}_{\geq 0}^M = \{x \in \mathbb{R}^M \mid x_m \geq 0 \text{ for all } m = 1, \dots, M\}$.

For period i , let $\mu_i \in \mathbb{R}^M$ denote the minimizer of (42) with $t_a := \tau_{i-1} + 1$ and $t_b := \tau_i$. We interpret μ_i as a vector of flows that suitably represents the turn movement flows throughout the i th period and is used to determine the signal timing parameters. For example, if

$$\Phi(x, \mu) = (x - \mu)^\top (x - \mu), \quad (44)$$

then the minimizing μ_i is given by $\mu_i = \frac{1}{\tau_i - \tau_{i-1}} \sum_{t=\tau_{i-1}+1}^{\tau_i} x(t)$, that is, μ_i is the average flow for the movements during period i .

Choosing $\Phi(x, \mu)$ as in (44) penalizes the difference between the turn movement flow and the corresponding value in the parameter vector. This choice equally penalizes turn movement flows that are above and below the parameter vector flow. However, it is often case that flows which exceed the parameter vector result in worse performance degradation at the intersection than flows which are less than the parameter vector. For this reason, we propose an asymmetric function

$$\Phi(x(t), \mu) = \sum_{m=1}^M \varphi(x_m(t), \mu_m) \quad (45)$$

$$\varphi(a, b) = \begin{cases} C(a - b)^2 & \text{if } a > b \\ (a - b)^2 & \text{else} \end{cases} \quad m = 1, \dots, M \quad (46)$$

for some choice $C \geq 1$ where μ_m denotes the m th entry of μ . For $C = 1$, we recover (44).

Finally, we choose segmentation times to minimize the sum of costs for all TOD periods:

$$(\tau_1, \dots, \tau_{S-1}) = \arg \min_{t_1, \dots, t_{S-1}} \sum_{i=1}^S \min_{\mu} F(\tau_{i-1} + 1, \tau_i, x, \mu). \quad (47)$$

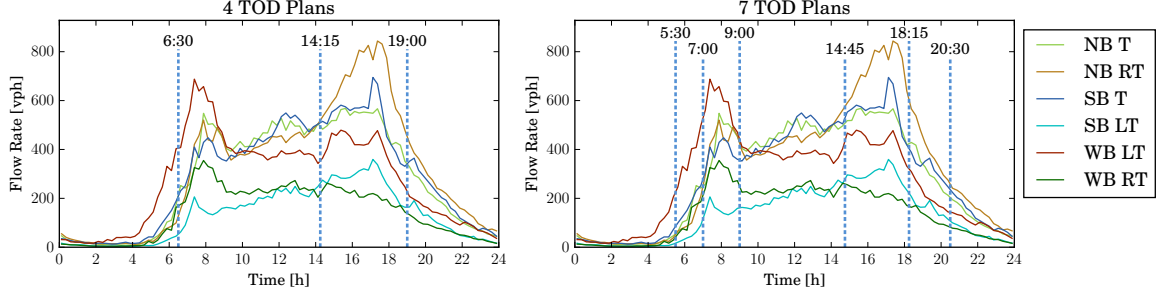


Figure 10: Optimal segmentation of average traffic flow over the day. (a) Optimal segmentation allowing for four time-of-day periods. (b) Optimal segmentation allowing for seven time-of-day periods.

Convexity of Φ ensures that (42) is a convex optimization problem. Assuming that solving (42) requires computational time of $\Gamma(M)$, then (47) is solved in time $O(M^2\Gamma(M))$ [33]. We note that our segmentation approach prevents oscillations in implemented timing plans since TOD periods are required to be contiguous intervals of time, in contrast to some clustering-based segmentation algorithms in the literature that do not account for contiguity in the TOD periods and for which the implemented signal timing plans have been observed to oscillate.

We return to our case study dataset and apply the TOD segmentation algorithm above where we take φ as defined in (46) with $C = 2$. We first take x to be the mean flow across the dataset as defined in (4)–(5), that is, we first consider $x := \bar{x}$ in (42)–(47). This constitutes the nominal TOD periods. In Figure 10, we plot the results of the optimal segmentation problem (47) with $S = 4$ and $S = 7$ for this nominal case. In general, we see that the segmentation algorithm chooses segmentation times corresponding to when turn movement flows are changing rapidly, resulting in intuitive divisions of the day. For example, with seven TOD periods, we clearly obtain the morning peak period from 7:00 to 9:00 and an afternoon peak period from 14:45 to 18:15.

5.3. Predictive Time-of-Day Plans

The segmentation algorithm presented in Section 5.2 where we use the mean flow \bar{x} is a method for optimally determining nominal TOD periods, an important component of conventional traffic control. Thus this is a useful innovation in itself that can be immediately implemented with existing hardware, but it does not leverage the traffic prediction algorithm developed in Section 4. Here we propose a method for extending this idea to allow real-time adjustments to TOD periods and TOD plans based on predicted traffic flow. The key idea is to enable limited and intuitive modifications to the standard TOD scheduling paradigm so that these methods can be implemented with modest modification to existing hardware, and traffic engineers are able to easily envision the benefits of traffic predictive control and therefore more easily adopt this approach.

To this end, we assume a set of nominal segmentation times $\tau_1, \dots, \tau_{S-1}$ have been chosen according to (47), defining a set of S TOD periods. Additionally, a set of parameter vectors μ_1, \dots, μ_S are obtained for each TOD period as the minimizers of (42). These TOD periods and parameter vectors constitute the nominal intersection timing plans.

We now suggest the following intuitive predictive traffic control scheme: throughout the day, online measurements are used to predict future traffic flow. When the current time approaches a nominal segmentation time, the predicted traffic flow is used to decide if the intersection controller should switch to a new TOD plan earlier or later than the nominal segmentation time.

To formalize this idea, let $x^s \in \mathbb{R}^{TM}$ denote the traffic flow for a particular sample day and suppose that the current time is t so that

$$x^{\text{meas}} := [x^s(1) \quad x^s(2) \quad \dots \quad x^s(t)] \in \mathbb{R}^{tM} \quad (48)$$

is the vector of currently available time measurements. We let

$$\text{SEGMENTWINDOW}(\tau) \subseteq \{1, 2, \dots, T\} \quad (49)$$

denote the window of times around a nominal segmentation time τ for which it is acceptable to switch early or late to a new TOD period. That is, the signal controller has the flexibility to switch from TOD period i to TOD period $i + 1$ at any time within $\text{SEGMENTWINDOW}(\tau_i)$ and the switch must occur within this window. The acceptable window is a design parameter, *e.g.*, it may be chosen by a traffic engineer to meet other system requirements. As an example, if the user-selected criterion allows for switching to a new TOD plan up to 45 minutes before or after the nominal segmentation time, we have $\text{SEGMENTWINDOW}(\tau) = \{\tau - 3, \tau - 2, \dots, \tau + 2, \tau + 3\}$ when $\Delta = 15$ minutes.

Now suppose that the current TOD period is i , the currently active parameter vector is $\mu_i^* \in \mathbb{R}^M$, and that $t \in \text{SEGMENTWINDOW}(\tau_i)$ so that it is allowable to switch to the $(i + 1)$ -th TOD period at the current time t . To determine if switching is desirable, we use a prediction of future traffic flow to determine if it is possible to achieve a lower segment cost by choosing a different segmentation time. To this end, let

$$\hat{y} = [\hat{y}(t + 1) \quad \hat{y}(t + 2) \quad \dots \quad \hat{y}(\tau_{i+1})] \in \mathbb{R}^{(\tau_{i+1} - t)M} \quad (50)$$

be a prediction of traffic flow from time $t + 1$ to time τ_{i+1} , the nominal ending time for the next TOD period. We wish to find the segmentation time which minimizes the predicted remaining segment cost for the i -th TOD period and the predicted segment cost for the $(i + 1)$ -th TOD period. We thus compute

$$t_{\text{opt}} = \underset{\substack{u \in \text{SEGMENTWINDOW}(\tau_i) \\ \text{s.t. } u \geq t}}{\arg \min} \left(F(t + 1, u, \hat{y}, \mu_i^*) + \min_{\mu} F(u + 1, \tau_{i+1}, \hat{y}, \mu) \right). \quad (51)$$

We interpret t_{opt} as the optimal time to switch from TOD period i to TOD period $i + 1$ (recall that for $u < t + 1$, we have $F(t + 1, u, \hat{y}, \mu_i^*) = 0$). If $t_{\text{opt}} = t$, then it is best to switch to the $(i + 1)$ -th TOD period immediately. In this case, we define

$$\tau_i^* = t_{\text{opt}} \quad (52)$$

$$\mu_{i+1}^* = \min_{\mu} F(u + 1, \tau_{i+1}, \hat{y}, \mu) \quad (53)$$

as the *predictive* segmentation time for the i -th period and the predictive parameter vector for the $(i + 1)$ -th TOD period. We then repeat the process for the next TOD segmentation time. However, if $t_{\text{opt}} > t$, then it is optimal to continue with the current TOD period, delaying the switch to the $(i + 1)$ -th period. Time advances to $t + 1$ and the process repeats, where we update the measurement vector and recompute the predicted flow measurements. In particular, this recomputation implies that t_{opt} updates based on the latest measurements. To initialize this approach, we define $\mu_1^* := \mu_1$, that is, the parameter vector for the first TOD period is assumed to be the nominal parameter vector μ_1 since no measurements are available yet that would alter this prediction.

Finally, we highlight one particularly important possible modification to the above procedure. In computing t_{opt} in (51), we assumed that we are able to establish a new parameter vector for the $(i + 1)$ -th TOD period, however due to constraints on the traffic signaling hardware or the preference of practitioners, this may not be possible. For example, the traffic signal controller may allow variability in the segmentation times defining TOD periods but not allow the timing plans themselves to be altered. In this case, we choose a new segmentation time assuming that the parameter vector remains fixed. We modify (51) as

$$t_{\text{opt}} = \underset{\substack{u \in \text{SEGMENTWINDOW}(\tau_i) \\ \text{s.t. } u \geq t}}{\arg \min} (F(t + 1, u, \hat{y}, \mu_i) + F(u + 1, \tau_{i+1}, \hat{y}, \mu_{i+1})). \quad (51b)$$

Algorithm 1 summarizes this traffic predictive approach. We now prove a desirable consistency property of this algorithm. In particular, we show that, for the fit function given in (46), if the predicted traffic flow is equal to the nominal traffic flow, then the segmentation times and parameter vectors obtained via Algorithm 1 are equal to the nominal segmentation times and parameter vectors. Below, we assume for simplicity that minimizing arguments are unique.

Proposition 1. Consider times $\tau_a < \tau_b < \tau_c$ defining two TOD periods $\{\tau_a + 1, \dots, \tau_b\}$ and $\{\tau_b + 1, \dots, \tau_c\}$, and consider the fit function given in (46). Assume that τ_b is the optimal time to switch from the first TOD period to the second TOD period when the flow is $x \in \mathbb{R}^{TM}$, and suppose μ_1 and μ_2 are the optimal parameter vectors for these periods, that is,

$$(\tau_b, \mu_1, \mu_2) = \arg \min_{\tilde{\tau}_b, \tilde{\mu}_1, \tilde{\mu}_2} F(\tau_a + 1, \tilde{\tau}_b, x, \tilde{\mu}_1) + F(\tilde{\tau}_b + 1, \tau_c, x, \tilde{\mu}_2). \quad (54)$$

Then, for all $t \in \{\tau_a, \dots, \tau_b\}$,

$$(\tau_b, \mu_2) = \arg \min_{\tilde{\tau}_b, \tilde{\mu}_2} F(t + 1, \tilde{\tau}_b, x, \mu_1) + F(\tilde{\tau}_b + 1, \tau_c, x, \tilde{\mu}_2). \quad (55)$$

Proposition 1 states that, if the measured data coincides with x , τ_b remains the optimal time to switch TOD periods even if the segmentation time and parameter vector of the second TOD is recomputed at time $t \geq \tau_a$.

Proof. Consider $t \in \{\tau_a, \dots, \tau_b\}$. Let

$$(\tau'_b, \mu'_2) = \arg \min_{\tilde{\tau}_b, \tilde{\mu}_2} F(t + 1, \tilde{\tau}_b, x, \mu_1) + F(\tilde{\tau}_b + 1, \tau_c, x, \tilde{\mu}_2). \quad (56)$$

Observe that

$$F(t_1, t_2, x, \mu) = F(t_1, t_3, x, \mu) + F(t_3 + 1, t_2, x, \mu) \quad \text{for all } t_1, t_2, t_3 \quad (57)$$

for any x, μ . Thus, by (54), we have that

$$F(\tau_a + 1, \tau_b, x, \mu_1) + F(\tau_b + 1, \tau_c, x, \mu_2) = F(\tau_a + 1, t, x, \mu_1) + F(t + 1, \tau_b, x, \mu_1) + F(\tau_b + 1, \tau_c, x, \mu_2) \quad (58)$$

$$\leq \min_{\tilde{\mu}_1, \tilde{\mu}_2} F(\tau_a + 1, \tau'_b, x, \tilde{\mu}_1) + F(\tau'_b + 1, \tau_c, x, \tilde{\mu}_2). \quad (59)$$

It follows that

$$F(t + 1, \tau_b, x, \mu_1) + F(\tau_b + 1, \tau_c, x, \mu_2) \quad (60)$$

$$\leq \min_{\tilde{\mu}_1, \tilde{\mu}_2} F(\tau_a + 1, \tau'_b, x, \tilde{\mu}_1) + F(\tau'_b + 1, \tau_c, x, \tilde{\mu}_2) - F(\tau_a + 1, t, x, \mu_1) \quad (61)$$

$$= \min_{\tilde{\mu}_1} F(\tau_a + 1, \tau'_b, x, \tilde{\mu}_1) - F(\tau_a + 1, t, x, \mu_1) + \min_{\tilde{\mu}_2} F(\tau'_b + 1, \tau_c, x, \tilde{\mu}_2) \quad (62)$$

$$\leq F(t + 1, \tau'_b, x, \mu_1) + \min_{\tilde{\mu}_2} F(\tau'_b + 1, \tau_c, x, \tilde{\mu}_2) \quad (63)$$

where the last inequality follows because

$$\min_{\tilde{\mu}_1} F(\tau_a + 1, \tau'_b, x, \tilde{\mu}_1) \leq F(\tau_a + 1, t, x, \mu_1) + F(t + 1, \tau'_b, x, \mu_1). \quad (64)$$

Then (55) follows from (56) and (60)–(63). \square

The form of the fit function given in (46) ensures (57) in the proof of Proposition 1. Thus, the proposition may fail to hold for alternative choices of fit functions.

Corollary 1. If $\hat{y} = [x^s(t + 1) \ x^s(t + 2) \ \dots \ x^s(\tau_{i+1})]$, that is, the predicted traffic flow is equal to the measured traffic flow, then $\tau_i = \tau_i^*$ and $\mu_i = \mu_i^*$ for all $i = 1, \dots, S$, that is, the predictive segmentation times and predictive parameter vectors are equal to the nominal segmentation times and parameter vectors.

```

1: function PREDICTIVETRAFFICCONTROL( $(\tau_1, \dots, \tau_{S-1})$ ,  $(\mu_1, \dots, \mu_S)$ ,  $x^s$ )
2:   inputs:  $(\tau_1, \dots, \tau_{S-1})$ , nominal segmentation times defining  $S$  TOD periods
3:              $(\mu^1, \dots, \mu^S)$ , parameter vectors for the TOD periods
4:              $x^s \in \mathbb{R}^{TM}$ , measured traffic flow, available in real-time
5:   outputs:  $(\tau_1^*, \dots, \tau_{S-1}^*)$ , predictive segmentation times
6:              $(\mu_1^*, \dots, \mu_S^*)$ , predictive parameter vectors
7:    $\mu_1^* := \mu_1$ 
8:    $i := 1$  ▷ Current TOD period
9:   for  $t = 1, 2, \dots, T$  do ▷ Current time
10:     $x^{\text{meas}} := [x^s(1) \ x^s(2) \ \dots \ x^s(t)] \in \mathbb{R}^{tM}$  ▷ Flow measurements up to current time
11:    if  $t \in \text{SEGMENTWINDOW}(\tau_i)$  then
12:       $\hat{y} = \text{PREDICT}(x^{\text{meas}}, t+1, \tau_{i+1})$  ▷ Predicted traffic flow up to end of next TOD period
13:       $t_{\text{opt}} := \text{according to (51) or (51b)}$  ▷ Best time to switch according to current prediction
14:      if  $t_{\text{opt}} = t$  then
15:         $\tau_i^* := t_{\text{opt}}$ 
16:         $\mu_{i+1}^* := \begin{cases} \arg \min_{\mu} F(\tau_i^* + 1, \tau_{i+1}, \hat{y}, \mu) & \text{if using (51)} \\ \mu_{i+1} & \text{if using (51b)} \end{cases}$ 
17:         $i := i + 1$ 
18:      end if
19:    end if
20:  end for
21:  return  $((\tau_1^*, \dots, \tau_{S-1}^*), (\mu_1^*, \dots, \mu_S^*))$ 
22: end function

23: function PREDICT( $x^{\text{meas}}$ ,  $t_a$ ,  $t_b$ )
24:   inputs:  $x^{\text{meas}} \in \mathbb{R}^{tM}$ , measured traffic flow up to time  $t$ 
25:              $t_a$  and  $t_b$ , start and end times for traffic flow prediction
26:   output:  $\hat{y} \in \mathbb{R}^{(t_b-t_a+1)M}$ , traffic flow prediction from time  $t_a$  to time  $t_b$ 
27:   summary: Using historical data and the prediction scheme developed in Section 4, predict traffic
28:             flow from time  $t_a$  to time  $t_b$  using real-time measurements  $x^{\text{meas}}$ .
29: end function

```

Algorithm 1: Algorithm for determining predictive segmentation times and predictive parameter vectors given a set of nominal segmentation times and vectors as well as a vector of current flow measurements. The algorithm runs online as time progress from $t = 1$ to $t = T$.

From a computational perspective, Algorithm 1 consists of an online component which computes the optimal predictive segmentation time for each TOD period as well as an offline component which processes historical data to obtain the PLS components for the predictions required in line 12. For the offline component, we compute PLS predictor/predicted components for each possible segmentation time. Thus, suppose $|\text{SEGMENTWINDOW}(\tau)| = W$ for all τ for some $W > 0$, that is, there are W allowable segmentation times around each nominal segmentation time. Then we execute the PLS algorithm a total of SW times. While this process is potentially computationally taxing, it can be accomplished offline by processing historical data. To obtain \hat{y} in line 12 of Algorithm 1 requires only matrix multiplication using the PLS components that are computed offline and stored. Next, computing t_{opt} requires solving at most W convex optimization problems if using (51), or W evaluations of the fit function F if using (51b). Solving the convex optimization problem is typically fast, as are evaluations of F , thus t_{opt} is easily computed within any practically sized time step Δ , affording ample time to decide predictive segmentation times and parameter vectors online.

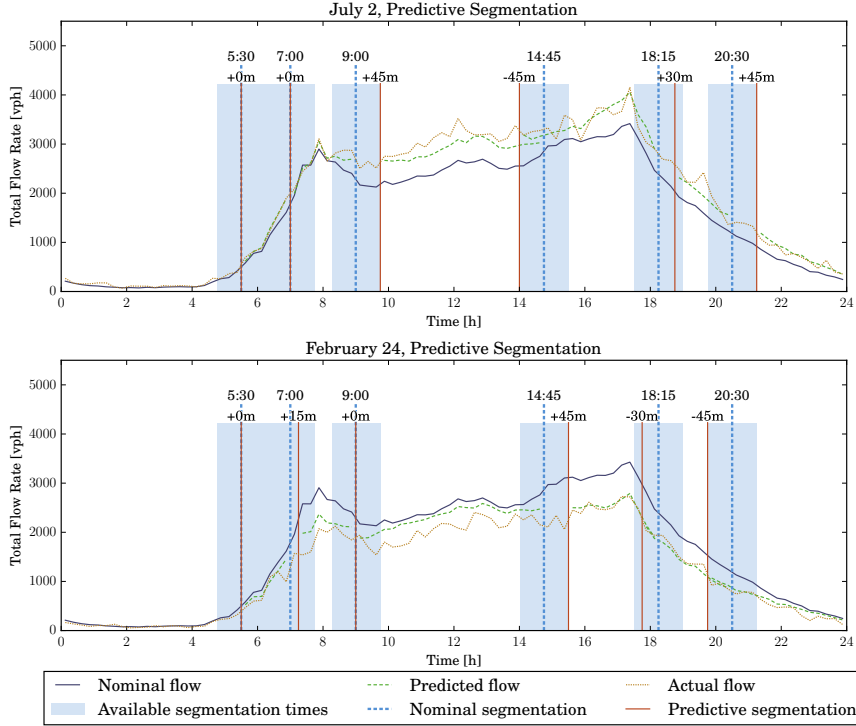


Figure 11: Example of traffic predictive control for two unusual days. The traffic predictive controller computes predictive segmentation times and adjusts the TOD periods to accommodate the predicted traffic flow, which deviates from the nominal average computed over the dataset. For clarity, the plots show total traffic flow through the intersection, summed over the twelve movements, however the prediction algorithm considers separate flow measurements for each movement.

5.4. Case Study

We assume nominal segmentation times for seven TOD plans as shown in Figure 10b. Now consider a traffic predictive controller as developed in Section 5.3 that is able to adjust in real-time the TOD periods so that the segmentation times are up to 45 mins earlier or later than the nominal segmentation times.

We first consider the case for which the traffic predictive controller is only able to establish predictive segmentation times and not predictive parameters. Figure 11 plots the results of executing the traffic predictive controller on the unusual dates February 24 (below average traffic due to weather conditions) and July 2 (above average flow due to holiday). The solid line in the figure depicts the mean *nominal* total traffic flow over the dataset (excluding the unusual date), summed over the 12 movements, in 15 minute intervals. Here, we plot total traffic flow only for clarity; the analysis considers separate flow measurements for all 12 movements as above. The dashed line is the total predicted traffic flow for the next TOD period as predicted at the start of each TOD period. The vertical dashed blue lines are the nominal segmentation times, the lightly shaded bars represent the segmentation windows during which a change to a new TOD period is allowed, and the solid vertical lines are the actual segmentation times as determined using the traffic predictive controller.

From the figure, it is evident that the traffic predictive controller behaves in a reasonable way. For example, on July 2, the nominal TOD period between 7:00 and 9:00 corresponds to the morning peak period. Since traffic is predicted to be above average at this time, this TOD period is extended by 45 minutes. Likewise, since traffic in the afternoon peak period is predicted to be above average, the corresponding TOD period is extended, beginning 45 minutes earlier and ending 30 minutes later. On February 24, we see the reverse phenomenon due to the prediction that traffic will be below average; the TOD periods for the morning and afternoon peak are correspondingly shortened.

A key feature of the traffic predictive controller as developed in Section 5.3 is that the scheme is agnostic regarding the specific green split algorithm utilized, that is, the algorithm seeks to predict flows and segmentation times, not green splits directly. However, to estimate the gain in performance, we must choose a particular green split algorithm. We employ a classical delay minimizing green split optimization algorithm as developed in [34], where we substitute modified delay formulas from the Highway Capacity Manual (HCM) [35]. These delay formulas account for queue buildup when the green splits are too short to accommodate queued vehicles. It is assumed that a fixed time control strategy is used within each TOD period calculated using the parameter vector, which is inflated by a factor to accommodate random fluctuations that would occur if vehicles arrive according to a Poisson process, a reasonable assumption.

In Figure 12, we plot the result of using this delay minimizing green split optimization with our traffic predictive controller. In each subplot, the solid trace is the rate of delay for the intersection plotted over time for the case where the green splits are designed to minimize delay using the nominal segmentation times and nominal parameter vectors. The rate of delay is calculated using the analytical formulas found in the HCM. The dashed trace is the rate of delay obtained using the traffic predictive controller. The top plots consider a traffic predictive controller that uses predictive segmentation times but nominal parameter vectors, that is, the green splits are not allowed to differ from nominal during each TOD period, however, the duration and starting point of the TOD period is adjusted based on the predicted flow. The bottom plots consider the case when the traffic predictive controller uses predictive segmentation times and predictive parameter vectors, thereby adjusting the green splits based on predicted traffic flow. The dotted trace indicates a lower bound on the rate of delay that is computed assuming that the optimal green splits for each fifteen minute interval are applied at each time step.

Suboptimality of green splits occurs due to two reasons: either the splits are too short for some movements, resulting in queued vehicles that must wait more than one cycle to clear the intersection; or the splits are too long, resulting in wasted green time whereby vehicles at other movements must wait longer than necessary to receive a green light. For the nominal control case, the former condition occurs on July 2 and the latter condition occurs on February 24. Indeed, the large increase in the rate of delay between 10:00 and 11:00 on July 2 results from queued vehicles waiting multiple cycles to move through the intersection, so-called *cycle failures*. These queued vehicles contribute substantially to the rate of delay at the intersection. Allowing predictive segmentation mitigates the issue somewhat as the traffic predictive controller delays by 45 minutes the change to a new TOD period that nominally occurs at 9:00, which in turn delays the onset of queued vehicles. However, since the nominal parameter vectors are used within each TOD period, the issue returns between 10:00 and 14:00. However, when predictive parameter vectors are considered as in the top right figure, leading to recomputed green splits for each TOD period, the rate of delay substantially decreases and is nearly equal to the lower bound.

On February 24, cycle failures are not a concern due to the below average traffic, however the nominal green splits lead to wasted green time. The rate of delay can be modestly improved by using predicted parameter vectors. The two trends exhibited by the conditions on July 2 and February 24 are general: cycle failures lead to large increases in the rate of delay while wasted green time typically affects delay by a lesser degree.

Total delay is computed by integrating rate of delay over the course of the day. Table 2 collects the total delay values for the four cases in Figure 12 as well as mean values taken over the entire data set. The traffic predictive controller reduces total delay at the intersection by up to 113.3 veh-hr on July 2 by mitigating excessive delay caused by cycle failures. This leads to a 22.1% decrease in total delay, a marked improvement especially considering that a lower bound on the best achievable delay reduction is 26.4%. Approximately 45 400 vehicles transited the intersection on this day, resulting in average savings of 9.0s per vehicle. Over the 132 days in the dataset, the traffic predictive controller that uses predictive segmentation and predictive parameters results in a delay improvement of 7.8 veh-hr on average, and the delay improvement exceeds 25 veh-hr on 11 occasions.

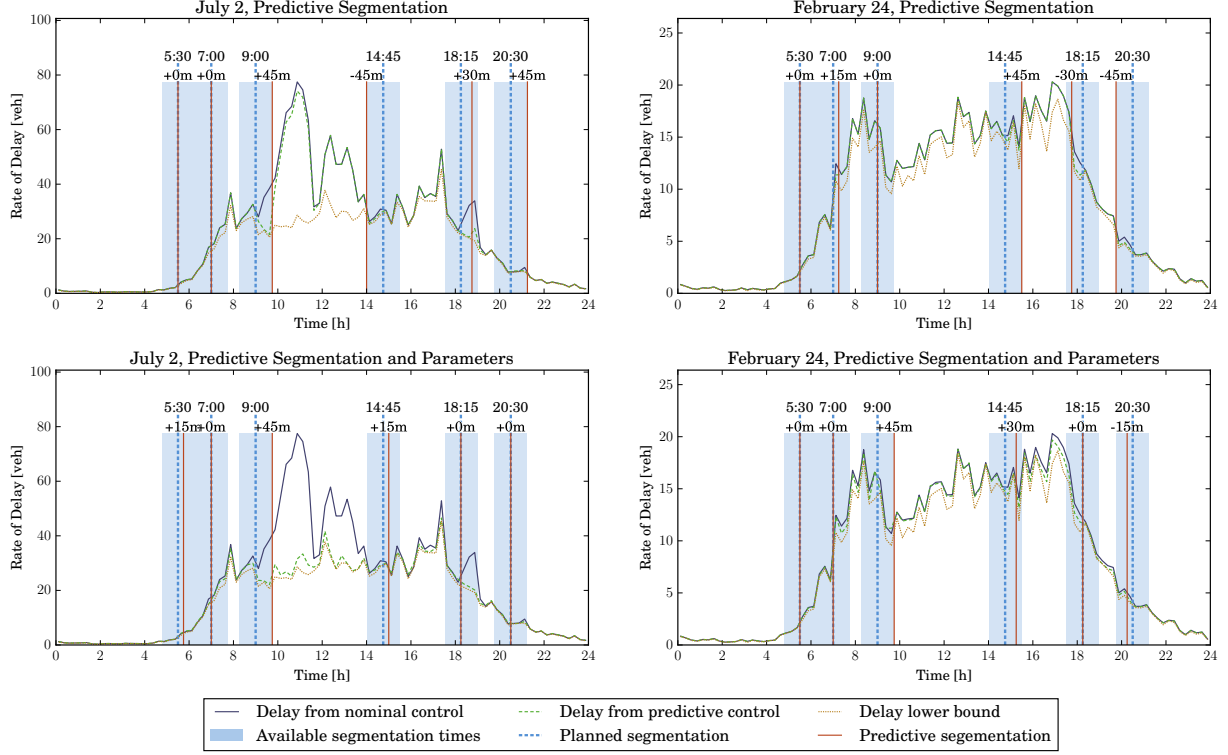


Figure 12: Rate of delay induced by the traffic predictive controller that adjusts TOD periods and TOD plans based on real-time traffic data as compared to the nominal controller that does not adjust TOD periods or plans. It is assumed that the green splits are computed to minimize delay using the parameter vectors for each TOD period. The top plots are the case when only predictive segmentation is considered while the bottom plots consider predictive segmentation and predictive parameter vectors. Traffic predictive control eliminates excessive delay caused by queued vehicles that require multiple cycles to clear the intersection and reduces wasted green time. Table 2 quantifies these delay improvements.

| | Feb 24 | July 2 | Dataset Mean |
|--|--------|--------|--------------|
| Delay, nominal control [veh·hr] | 209.1 | 512.1 | 305.7 |
| Delay, predictive segmentation [veh·hr] | 207.5 | 490.5 | 303.1 |
| Delay, predictive segmentation and parameters [veh·hr] | 204.2 | 398.8 | 297.9 |
| Delay lower bd. [veh·hr] | 192.5 | 377.5 | 276.4 |
| Delay improvement, predictive segmentation [veh·hr] | 1.7 | 21.6 | 2.6 |
| Delay improvement, predictive seg. and param. [veh·hr] | 5.0 | 113.3 | 7.8 |

Table 2: Illustrative delay savings from using traffic predictive control to adjust TOD periods and plans. The traffic predictive controller reduces total delay at the intersection by 113.3 veh·hr on July 2 by mitigating excessive delay caused by queued vehicles that wait multiple cycles to clear the intersection, so-called cycle failures. On average, the traffic predictive controller improves total delay by 7.8 veh·hr per day by preventing cycle failures and reducing wasted green time.

6. Conclusions

We have proposed a traffic predictive control scheme that identifies trends in historical traffic flow data and uses real-time measurements to predict future traffic flow. These trends manifest as low-rank structure in the data which are identified using decomposition techniques akin to principal component analysis and particularly suited for prediction. Using a rich dataset of traffic flow measurements over the course of eight months, we provide evidence that much of the day-to-day variation in traffic flow consists of these low-rank, latent structures.

The traffic predictive control adjusts the time periods and parameters for time-of-day traffic signal scheduling based on predictions of traffic flow using real-time measurements. This scheme is particularly well-suited for implementation on existing traffic control hardware, which universally support time-of-day plans and often are capable of remote updating of signal timing parameters. Furthermore, this approach is well-aligned with standard signal timing practices, increasing the likelihood of successful adoption by practitioners. Additionally, the traffic predictive control requires minimal tuning and accommodates any green split optimization scheme that requires expected traffic flow as input.

The savings in delay for the case study intersection is found to be 7.8 veh-hr per day. Valuing a driver's time at \$20 per hour and conservatively assuming that each vehicle carries only one occupant, this suggests annual savings of around \$57 000. In addition, this metric is compared to a well-timed but pre-specified controller that does not account for real-time measurements; the savings are likely to be higher for intersections that are currently poorly timed. If these calculations are even only approximately correct, the savings are likely to be well worth the relatively small implementation costs.

An important future direction of research is to consider the case of an arterial corridor. In this case, a straightforward extension of the proposed approach is to consider all movements along the corridor in aggregate. However, there is high correlation between the traffic flows for some sets of movements as vehicles progress along the corridor. This suggests an approach that explicitly accounts for these spatial correlations as is done in [36].

Additionally, future research will investigate confidence bounds on the predictions. For example, can low-rank structure be used to predict the 90th or the 99th percentile traffic flow? Furthermore, traffic prediction from historical trends captures phenomena that have occurred previously in the historical data, and thus may not be well-suited for events that are truly one-off events such as lane closures due to construction. However, it may be that lane closures in one direction over an interval of time affects traffic in a similar manner as lane closures on a different leg at a different time. Learning these more universal trends is another future direction of research. Finally, these questions will be pursued alongside an implementation pilot planned in collaboration with Sensys Networks, Inc.

Acknowledgements

The authors acknowledge Beaufort County, SC for use of the intersection data and thank Montasir Abbas for discussions regarding technical capabilities of common traffic control hardware and best practices for signal timing and plan selection.

References

References

- [1] A. A. Kurzhanskiy and P. Varaiya, "Traffic management: An outlook," *Economics of Transportation*, vol. 4, no. 3, pp. 135–146, 2015.
- [2] R. Dowling and S. Ashiabor, "Traffic signal analysis with varying demands and capacities, draft final report," Tech. Rep. NCHRP 03-97, Transportation Research Board, 2012.
- [3] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 annual urban mobility scorecard," 2015. <http://mobility.tamu.edu/ums/report/>.
- [4] P. Koonce, L. Rodegerdts, K. Lee, S. Quayle, S. Beaird, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik, "Traffic signal timing manual," tech. rep., U.S. Department of Transportation, Federal Highway Administration, 2008.

- [5] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport reviews*, vol. 24, no. 5, pp. 533–557, 2004.
- [6] S. E. Jabari and H. X. Liu, "A stochastic model of traffic flow: Theoretical foundations," *Transportation Research Part B: Methodological*, vol. 46, no. 1, pp. 156–174, 2012.
- [7] S. E. Jabari and H. X. Liu, "A stochastic model of traffic flow: Gaussian approximation and estimation," *Transportation Research Part B: Methodological*, vol. 47, pp. 15–41, 2013.
- [8] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive Kalman filtering for multi-step ahead traffic flow prediction," in *American Control Conference (ACC)*, 2013, pp. 4724–4729, IEEE, 2013.
- [9] C.-J. Wu, T. Schreiter, and R. Horowitz, "Multiple-clustering ARMAX-based predictor and its application to freeway traffic flow prediction," in *American Control Conference (ACC)*, 2014, pp. 4397–4403, IEEE, 2014.
- [10] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using Bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008.
- [11] X. Zhou and H. S. Mahmassani, "A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework," *Transportation Research Part B: Methodological*, vol. 41, no. 8, pp. 823–840, 2007.
- [12] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1097–1122, 2012.
- [13] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transportation Research Part B: Methodological*, vol. 53, pp. 64–81, 2013.
- [14] L. Du, S. Peeta, and Y. H. Kim, "An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks," *Transportation Research Part B: Methodological*, vol. 46, no. 1, pp. 235–252, 2012.
- [15] C. Osorio, G. Flötteröd, and M. Bierlaire, "Dynamic network loading: a stochastic differentiable model that derives link state distributions," *Transportation Research Part B: Methodological*, vol. 45, no. 9, pp. 1410–1423, 2011.
- [16] I. Guardiola, T. Leon, and F. Mallor, "A functional approach to monitor and recognize patterns of daily traffic profiles," *Transportation Research Part B: Methodological*, vol. 65, pp. 119–136, 2014.
- [17] P. Mirchandani and L. Head, "A real-time traffic signal control system: architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.
- [18] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*, pp. 34–51, Springer, 2006.
- [19] M. Abbas and A. Sharma, "Multiobjective plan selection optimization for traffic responsive control," *Journal of Transportation Engineering*, vol. 132, no. 5, pp. 376–384, 2006.
- [20] A. Haoui, R. Kavalier, and P. Varaiya, "Wireless magnetic sensors for traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 294–306, 2008. Emerging Commercial Technologies.
- [21] A. Muralidharan, S. Coogan, C. Flores, and P. Varaiya, "Management of intersections with multi-modal high-resolution data," *Transportation Research Part C*, vol. 68, pp. 101–112, 2016.
- [22] S. Wold, A. Ruhe, H. Wold, and W. Dunn, III, "The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 735–743, 1984.
- [23] P. Geladi and B. R. Kowalski, "Partial least-squares regression: A tutorial," *Analytica chimica acta*, vol. 185, pp. 1–17, 1986.
- [24] R. Manne, "Analysis of two partial-least-squares algorithms for multivariate calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 187–197, 1987.
- [25] A. Höskuldsson, "PLS regression methods," *Journal of Chemometrics*, vol. 2, no. 3, pp. 211–228, 1988.
- [26] H. Abdi, "Partial least squares regression and projection on latent structure regression (PLS regression)," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 97–106, 2010.
- [27] S. Rännar, F. Lindgren, P. Geladi, and S. Wold, "A PLS kernel algorithm for data sets with many variables and fewer objects. Part 1: Theory and algorithm," *Journal of Chemometrics*, vol. 8, no. 2, pp. 111–125, 1994.
- [28] F. V. Webster, "Traffic signal settings," tech. rep., 1958.
- [29] B. L. Smith, W. T. Scherer, T. A. Hauser, and B. B. Park, "Data-driven methodology for signal timing plan development: A computational approach," *Computer-Aided Civil and Infrastructure Engineering*, vol. 17, no. 6, pp. 387–395, 2002.
- [30] R. Guo and Y. Zhang, "Identifying time-of-day breakpoints based on nonintrusive data collection platforms," *Journal of Intelligent Transportation Systems*, vol. 18, no. 2, pp. 164–174, 2014.
- [31] B. Park, D.-H. Lee, and I. Yun, "Enhancement of time of day based traffic signal control," in *IEEE International Conference on Systems, Man and Cybernetics, 2003.*, vol. 4, pp. 3619–3624, Oct 2003.
- [32] B. B. Park, P. Santra, I. Yun, and D.-H. Lee, "Optimization of time-of-day breakpoints for better traffic signal control," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1867, pp. 217–223, 2004.
- [33] I. E. Auger and C. E. Lawrence, "Algorithms for the optimal identification of segment neighborhoods," *Bulletin of Mathematical Biology*, vol. 51, no. 1, pp. 39–54, 1989.
- [34] R. E. Allsop, "Delay-minimizing settings for fixed-time traffic signals at a single road junction," *IMA Journal of Applied Mathematics*, vol. 8, no. 2, pp. 164–185, 1971.
- [35] Transportation Research Board, "Highway capacity manual," 2000.
- [36] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.